International Advanced Diploma in Computer Studies

Business Systems Analysis

inspiring your success

Business Systems Analysis

Published by:
NCC Education Limited
The Towers, Towers Business Park
Wilmslow Road, Didsbury
Manchester M20 2EZ, UK

Tel: +44 (0) 161 438 6200
Fax: +44 (0) 161 438 6240

email: enquiries@nccedu.com
http://www.nccedu.com

ISBN 1-90234-348-4

First edition published 2002

Typeset in 10 on 12pt Palatino
Printed by B & Jo Enterprises PTE in Singapore

NCC Education Portfolio

Our portfolio includes comprehensive course packages, competence testing and accreditation services.

Academic Awards

- English Language Framework (ELF)
- International Foundation Year (IFY)
- Computer Pioneers (CP)
- International Certificate in Computer Studies (ICCS)
- International Diploma in Computer Studies (IDCS)
- International Advanced Diploma in Computer Studies (IADCS)
- International Diploma in eCommerce (IDEC)
- BSc (Hons) in Computing and Information Systems (BSc CIS) – awarded by London Metropolitan University
- International Diploma in Business (IDB)
- International Advanced Diploma in Business (IADB)
- Postgraduate Diploma in Strategic Business Information Technology (PgD SBIT)
- MSc in Strategic Business IT with the University of Portsmouth (MSc SBIT)
- MBA in International Business awarded by Salem International University

Professional Programmes

- A+
- eCommerce
- Multimedia
- Developing a Website
- C++
- VB.NET
- Java
- Professional Development Diploma in ICT for Teachers
- TechMaster Courses
- Systems Analysis Certification
- Systems Design Certification
- Database Design and Development
- Project Management
- Business Management

Products

- Student materials supporting NCC Education courses
- Lecturer materials to support our courses
- Automated Test Software for European Computer Driving Licence (ECDL/ICDL)
- ATS – Automated Test Software, including bespoke versions
- NCC Education PC Competency Test
- NCC Education Skills Profile Test
- NCC Education Recruitment Test

Services

- Course Accreditation
- Centre Accreditation

NCC Education

NCC Education was originally the education division of the National Computing Centre. The National Computing Centre was set up by the UK government in 1966 to support and promote the use of computing technology within industry. As such, it developed strong links with the Information Technology (IT) industry.

NCC Education Limited became a separate company in April 1997, since when we have introduced many more IT education programmes to our portfolio. Our mission is to ensure the widespread availability of quality education and training for developers and users of IT.

Experts in our Field

NCC Education is one of the world's leading IT qualification awarding bodies, with numerous academic and professional educational and training courses being taught in over 30 countries.

NCC Education's specialists work with government and national bodies worldwide to devise and implement educational and training projects. NCC Education's training programmes are recognised internationally as being an effective means for students to advance careers in IT.

NCC Education provides comprehensive support from the outset, from the provision of comprehensive tutor and student training materials to moderating the process of training delivery and establishing the assessment mechanism.

Who are our Partners?

Our unique position bridging IT education and industry allows us to develop partnerships within many different types of organisation. In addition to our work with government and national bodies, our academic programmes are run in IT and business colleges worldwide, our professional programmes are run in colleges, large companies and IT training organisations worldwide and our automated competence tests are run in companies and IT training organisations worldwide.

ISO 9001: 2000 Quality in Education

Historically, our aim has been to equip the UK with personnel who are competent in using and developing IT in business. As such, quality assurance is a rigorous requirement of all NCC Education products and partners.

NCC Education has gained, and continuously maintains, a quality standard throughout the organisation. Its certificate number 3812/03 was awarded by ISOQAR. The NCC Education Quality manual states:

> *NCC Education is unreservedly committed to satisfying its customers through quality in all NCC Education's products and services.*

In support of this policy, NCC Education operates a Quality System that provides a framework of standards and procedures within which it manages and controls all its project, product and service activities.

The Quality System is based on the requirement of the pertinent parts of the ISO 9001: 2000, independently assessed for compliance to the appropriate standard.

Specifically NCC Education's quality objectives are:

- To provide the highest quality administration and support to the international qualifications allied certification schemes.
- To ensure that NCC Education accredited centres provide courses that meet the requirements specified in the NCC Education syllabus and regulations.
- To specify and maintain syllabi and regulations that meet the career development needs of users and computer specialists in the IT community.
- To ensure where appropriate that syllabi meet national academic standards.

026

Certificate No: 3812/03

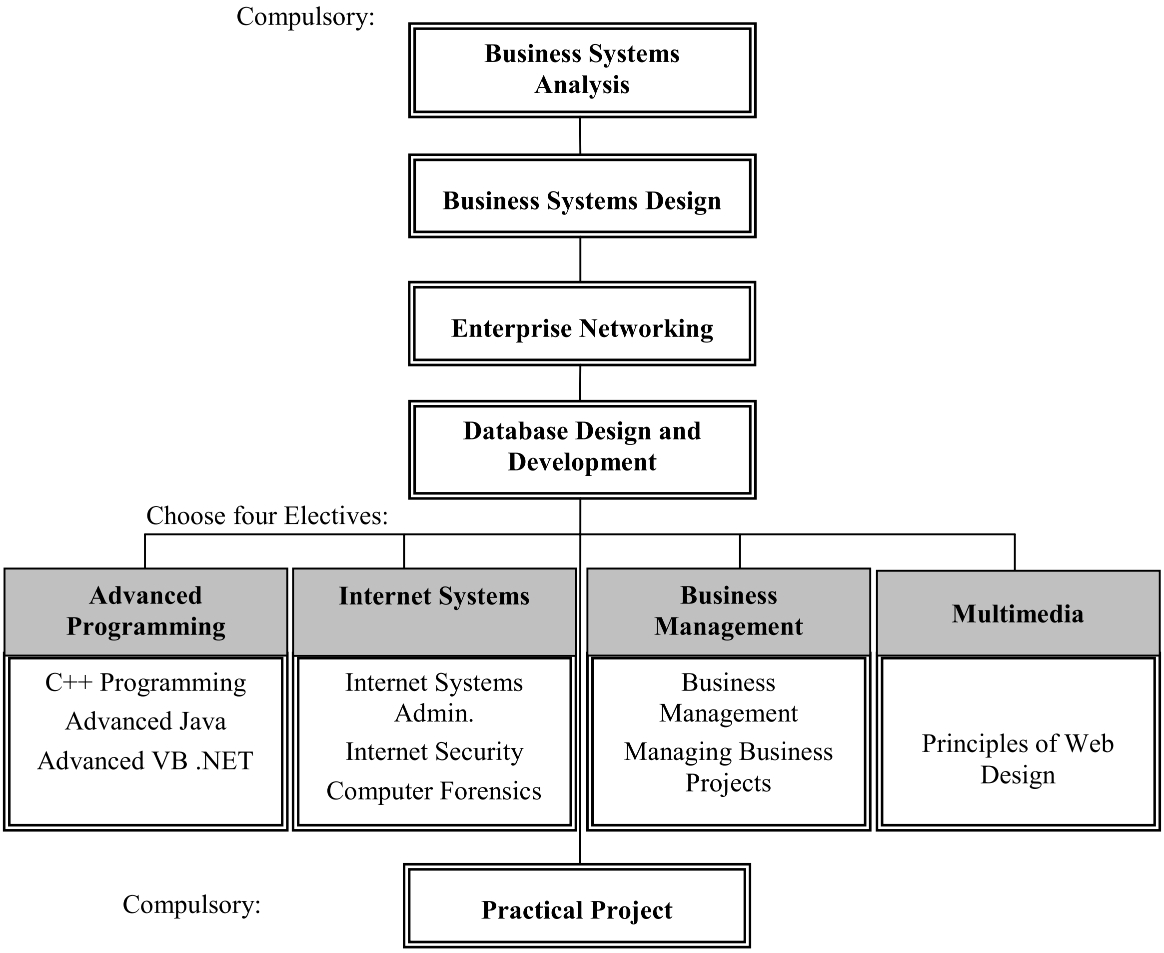
International Advanced Diploma in Computer Studies

The IADCS qualification is part of the NCC Education academic pathway. It is an intermediate level academic programme for people who wish to enhance their basic knowledge of computer studies.

The IADCS gives candidates the ability to ‘pick and mix’ subjects, allowing them the freedom to explore the topics and technologies which will be of most benefit in developing their career.

The IADCS is a one-year programme of full time study, comprising of four compulsory modules, four elective modules and one practical project. Time is divided equally between lectures, practical exercises and self-study.

NCC Education carries out time constrained written examinations for the compulsory modules, with the whole process being managed from the United Kingdom. Accredited Centres hold responsibility for assessing the elective modules using clear instructions from NCC Education as to how assessment is to be carried out. Assessment of each module may be by assignment, practical project, examination or a combination of these.



Contents

1
Planning and preparation

Introduction

The work of the systems analyst needs to be seen in the context of the whole systems life cycle, from initiation of the project to its implementation and beyond through maintenance. It also needs to be placed in the context of the way in which the business or organisation operates. The early chapters of this book address the organisation of business and systems within it.

The systems life cycle consists, in essence, of a series of phases: requirements analysis and specification, analysis, design, code and unit test, testing and implementation. The system life cycle is discussed in detail in a later chapter.

Discussion of the stages in system development (the life cycle) is followed by detailed discussion of the techniques which are available to the systems analyst, to assist in the analysis and documentation of the new system requirements, such as:

- Data Flow Diagrams (DFDs);
- Data Dictionary (DD);
- Miniature Specifications (mini-specs);
- Entity Analysis;
- Relational Data Analysis;
- Entity Life Histories (ELHs).

These techniques take different views of the system and provide a means of cross-checking the analyst's findings to identify anything which may have been missed.

This chapter sets the scene of the analyst's task by considering the purpose of analysis, and the planning for testing and implementation which must be

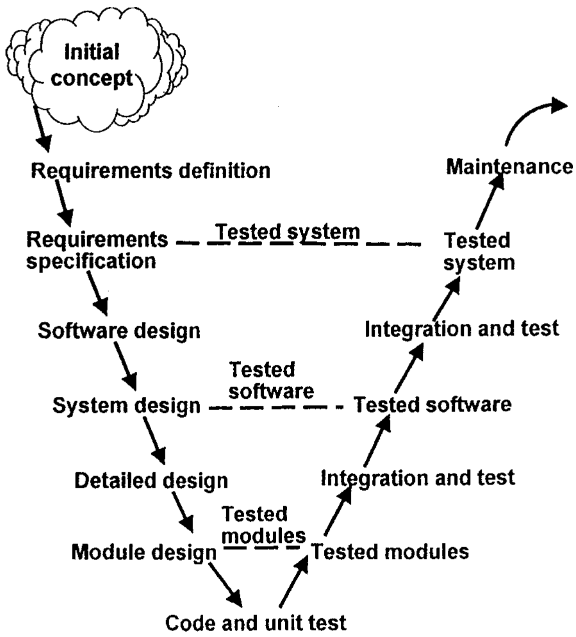


Figure 1.1 V-Model.

given consideration at the outset of a project. To facilitate the discussion of test planning in this chapter we shall take a preview of the V-Model form of the system life cycle which shows the relationship between the stages of system development and the test processes. The V-Model is shown in Figure 1.1.

1.1 The purpose of systems analysis

Systems analysis refers to those activities undertaken *to produce a System Specification* which will then be used as the basis for system design and build.

The system specification, which the analyst produces, is a clear and unambiguous statement of what the new system will do. It clearly defines the system in terms of the business requirements for that system and provides a logical, detailed description of what the system will do. The specification is agreed with the system users before design commences. Where an incremental approach to development is taken (as in a Rapid Application Development (RAD) approach, it is still important to analyse each increment of the system before it is designed and built. Where an existing system is in place examination of the current system and its shortcomings form a useful starting point. Sometimes there will not be an existing system and requirements are

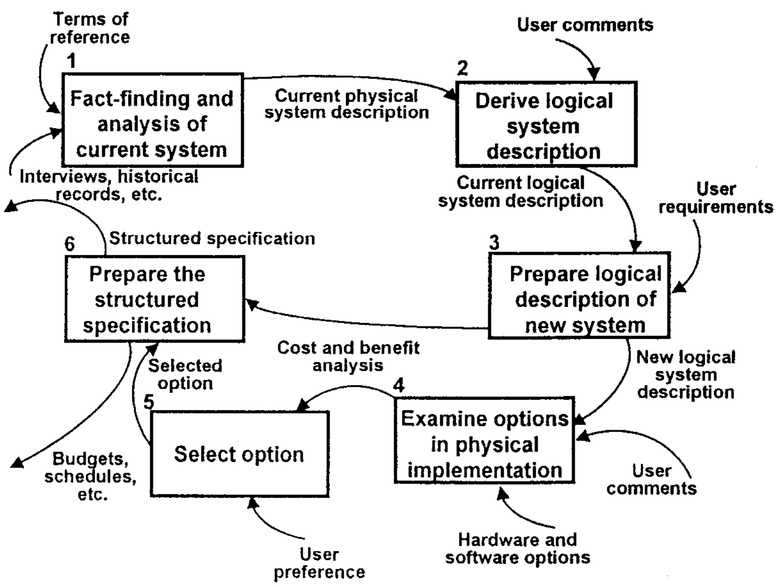


Figure 1.2 Stages in analysis.

more difficult to identify – it is always easier to discuss what is needed by reference to something which is already in place.

Figure 1.2 describes the process of analysis in a general way without reference to any particular method for systems analysis. This diagram will appear from time to time throughout this book.

1.2 Planning is essential

A project for development of software needs to be carefully planned from the outset, and a project initiation statement (document) should be raised to state the project scope and objectives, constraints and resources. This is discussed later in conjunction with terms of reference for the systems analyst. Experience suggests that the most practical approach to ensuring that planning and control takes place is the formation of a project steering committee. Such a committee is formed at an early stage of the project and is responsible for monitoring the project's progress throughout the analysis and design phases. The control of specific tasks is often delegated to a subset of this committee.

A steering committee may be a mixed blessing. Whilst it allows problems to be discussed and resolved by involving all interested parties, it can lengthen the decision-making process, and its success rate is probably determined largely by the personality and rank of its chairman and the careful choice of its members. The chairman should either have executive authority himself (as the 'executive sponsor' with ultimate responsiboility for the project) or have easy access to top management to eliminate any potential delays in the decision-making process. The user departments affected must be represented both by management and, for matters concerning the specific details of jobs, by more junior staff. It is often appropriate to invite trade union participation. The IT Department will contribute by supplying the IT Manager and systems analysts. Meetings should be held regularly and actions and decisions should be carefully recorded.

In addition to the process of software development, planning should encompass a number of elements related to the introduction of the new system:

- ❏ Any **additional infrastructure requirements** to support the new system. Preparation of the offices and buildings to receive the new system needs to be planned and carried out before the system can go live. New equipment needs to be installed and tested before system trials can start;
- ❏ Any **staff issues** introduced by the new system. Any change in job roles need to be planned for. Training and motivation of users is of prime importance to create a climate in which the system will be readily accepted. Such training in the background of the system together with well-written user manuals will facilitate efficient use of the system;
- ❏ **Testing** needs to be planned for and carried out throughout the project. The typical life cycle model implies that testing is at the end of the system life cycle. Some testing does take place late in the development process but much of the testing is interleaved with the development of prototypes (where used), modules and program units;
- ❏ **Implementation:** the process of cut-over from the old system to the new one must be carefully planned. This should include planning for the transfer and conversion of data and the involvement of staff. It may be necessary to plan for a period of parallel running of the old and new systems, with the additional resources which this implies.

1.3 The analyst's focus on testing

Whilst systems analysis progresses and the business requirements are being defined for compilation into the system specification, it is the analyst's responsibility to also consider what testing is going to be carried out and when. The specification defines what the system is going to do and as this definition is built up and fresh in the analyst's mind, he/she should plan what tests will be used to confirm that the system does what the specification says that it should do. '*Say what you are going to do, do it and prove that you have done it*" is the rationale to be applied.

1.3.1 Verification and validation

Testing is about verification and validation: verification is the process of evaluating that the system does what it was specified to do. Validation is the testing that the system does what is really required for the business.

1.3.2 The high level test plan

At the initiation of a project, plans for testing should be documented. This document, the high-level test plan (or master test plan) must define the scope, approach and responsibilities for testing that need to be planned at the start of the project. The typical contents of a high level test plan are described below:

- ❑ **Introduction** This should cover the contents of the test plan plus references to other relevant documents such as Quality Assurance and Configuration Management plans.
- ❑ **Test Items** This should identify the physical things that are to be tested such as executable programs, data files or databases, with details of how they will be handed over to testing.
- ❑ **Features to be Tested/ not to be Tested** The functionality and features to be tested/ not to be tested.
- ❑ **Approach** The activities necessary to carry out the testing, the techniques and tools to be used and the completion criteria and constraints.
- ❑ **Pass / Fail Criteria** For each test item the criteria for passing or failing that item, such as the number and criticality of known outstanding faults.
- ❑ **Suspension / Resumption Criteria** The criteria to be used to determine when any testing activities should be suspended/resumed. For example, if too many faults are found with the first few tests, it may be more cost effective to stop testing at the current level and wait for the faults to be fixed.
- ❑ **Test Deliverables** What the testing process must provide in terms of documents, reports, etc.
- ❑ **Testing Tasks** Specific tasks, special skills required and the inter-dependencies.
- ❑ **Environment** Details of the hardware and software that will be needed in order to execute the tests. and any other facilities (including office space and desks) that may be required.
- ❑ **Responsibilities and Approvals** Who is responsible for which activities and deliverables.
- ❑ **Staffing and Training Needs** Staff required and any training they will need.
- ❑ **Schedule** Milestones for delivery of software into testing, availability of the environment and test deliverables.
- ❑ **Risks and Contingencies** What could go wrong and what will be done about it to minimise adverse impacts if anything does go wrong.

1.3.3 Stages of testing and the V-Model

In the V-Model of the system life cycle (Figure 1.1), the test activities parallel the analysis, design and coding (build) activities. Software is designed on the left-hand (downhill) part of the model, and built and tested on the right-hand (uphill) part of the model. The correspondences between the left and right hand activities are shown by the lines across the middle of the V, showing the test levels from component testing at the bottom, integration and system testing, and acceptance testing at the top level.

Tested modules (Component testing): a component is 'A minimal software item for which a separate specification is available' (BS7925). Components are small pieces of software, the building blocks from which the system is formed. They are also referred to as modules, units or programs and so this level of testing may also be known as module, unit or program testing. The programmer who wrote the code most often performs component testing.

Integration and test (1) (Integration testing in the small): this brings together individual components (modules/units) that have already been tested in isolation. The objective is to test that the 'set' of components function together correctly by concentrating on the interfaces between the components. This is an important level of testing but one that is, sadly, often overlooked.

Tested software (System testing): this stage has two important aspects: functional system testing (testing business processes and requirements) and non-functional system testing (testing such qualities as performance, response time, usability, load, performance and stress testing). The non-functional aspects are often as important as the functional, but are generally less well specified and may therefore be more difficult to test (but not impossible).

Integration and test (2) (Integration Testing in the large): this is concerned with the testing of the system with other systems and networks.

Tested system (Acceptance Testing): user acceptance testing is the final stage of validation and the end product of this is usually a sign-off from the users. One of the problems is that this is rather late in the project for users to be involved – any problems found now are too late to do anything about them. This is one reason why Rapid Application Development (RAD) has become popular – users are involved earlier and testing is done earlier. The users should, in any event, be involved in the test specification of the Acceptance Tests at the start of the project, and in reviews throughout the project.

If a system is the subject of a legally binding contract, there may be aspects of acceptance testing directly related to the contract. It is important to ensure that the contractual documents are kept up to date; otherwise the systems developers may be in breach of the contract while delivering what the users want (instead of what they specified in the contract).

A more detailed discussion of testing is to be found in the book: 'Business Systems Design'. It should be noted that system test data and its results should be preserved as part of the system documentation throughout the life of the system. A means of recording and retrieving such information has to be planned early in the development of the new system.

1.4 Analysis tasks to assist in the planning process

During the early stages of the project, the analyst must find out about the environment into which the new system will be introduced. To assist with this process, the following is a checklist of the areas about which the analyst must question.

Planning the design methods

- ❏ identify the potential users of the proposed system;
- ❏ interview users to establish what experience they have had of systems development. Use the interviews to become familiar with the work environment and establish a rapport with the users;
- ❏ decide on a particular method to obtain constructive user contributions to the design, bearing in mind the benefits and problems involved;
- ❏ plan any necessary training to allow users to follow the chosen method;
- ❏ inform all parties of the method that is to be employed and ensure that they understand and agree with the roles they will have to play.

Planning for disruption

- ❏ identify any disruptions likely to occur in the following categories:
  - a. user contributions to systems design;
  - b. physical changes to the workplace;
  - c. training users to operate the new system;
  - d. systems testing, parallel running and system initiation.
- ❏ plan to cater for the disruptions.

Work situation appraisal

- ❏ become familiar with the user's work environment, gain an appreciation of the nature of the work, the pressures, priorities, skills, values and attitudes of the people;
- ❏ identify the specific human factors information to be collected.

User identification

- ❑ identify potential users by identifying the name and job titles of the users of the existing system;
- ❑ develop a 'map' of the user population. For example, it will be helpful to classify future users in terms of primary, secondary or tertiary users to avoid focusing solely upon the main users at the expense of others;
- ❑ use the 'map' as a directory of potential users when user needs and characteristics have to be established at various points in the development.

Task demands

- ❑ identify the range and functions undertaken by the users with particular emphasis on the variations in the work;
- ❑ review techniques for task analysis and select appropriate techniques;
- ❑ conduct detailed analyses of user tasks with emphasis on task differences.

Work role analysis

- ❑ gain an understanding of the user's perception of their work, its purpose, value and meaning for them;
- ❑ gain the user's views on the rewards and costs of the job;
- ❑ identify the most important aspects of the job for the user.

User readiness

- ❑ assess users' previous experience of computing;
- ❑ examine users' attitude to computer use;
- ❑ assess user knowledge of the proposed system;
- ❑ assess users' willingness to cooperate in system development.

Job design

- ❑ identify all tasks, old and new, which will undergo change as a result of the new system;
- ❑ generate alternative ways of allocating tasks to jobs by considering, for example, allocations by function, product or customer;
- ❑ provide opportunities for user groups to explore job tasks by providing prototypes of the main functions of the new system.

Design methods for faults and breakdowns

- ❑ design methods such that users can take positive action when they are confronted by a fault or breakdown. Ensure that users are trained in these methods;

❑ ask users for their constructive criticism of the above methods which will affect them. Produce tests for faults and breakdowns.

Asking these questions at an early stage in the project allows the risks of the project to be better assessed and controlled, in addition to providing essential information for the planning of implementation.

1.5 Summary

Whilst gathering information during the Analysis phase of the system life cycle, it is essential for the analyst to consider what information and planning is needed in order to test whether the system meets the business requirements that underpin its development. In analysis we are not merely concerned with what the new system will do but also we must prove that we have developed a system that will do what the business wants it to do. Losing track of the business purpose with a focus only upon what the system will do will lead to technically correct programs that fail to meet the business need and therefore fall into disuse.

The project must be planned from the outset and be clear in its objectives and scope. Testing must be planned into the development process during the very early stages of a project to ensure that the right product is developed and accepted by the users of the system when it is eventually implemented. The V-Model of the system life cycle has been introduced here with further discussion in a later chapter. Its inclusion at this point is to provide a structure upon which the planning of the testing can be visualised.

Exercises

1.1 What is the purpose of systems analysis?
1.2 What is a steering committee? Who would be represented on one of these?
1.3 What is meant by verification and validation?
1.4 What is the V model?
1.5 What are the contents of a high level test plan?
1.6 What sort of information should the analyst be collecting regarding the identification of work roles and job design?

2

The organisation of business and information within business

Introduction

The systems analyst does not work in isolation. He/she is very much part of a team, both in relation to a particular project, and as a part of the successful operation of the whole business or organisation (enterprise). Indeed, the systems analyst's role is often combined with than of 'business analyst'. The responsibilities of the business analyst include not just the specification of a computer system, but the definition of a business system, with business processes and requirements, of which a part may require technology support (the computer system). The analyst must become familiar with the enterprise and understand its objectives in order to produce cost effective systems.
This chapter is a start to providing a general understanding of business. It seeks to answer the following questions:

- what types of business are there? (e.g. what commodities do they deal in?);
- what different types of enterprise are needed to run them and how are they financed?
- what goes on inside them? (e.g. what functions are carried out in a typical business enterprise?);
- what are their objectives?
- what kind of organisation structures are used to co-ordinate them?
- what is the role of management in a typical business enterprise?
- what is the role of data and information within a business?

2.1 Types of business

A glance at basic human needs shows us how business has evolved. In primitive societies, food gathering, hunting, and subsequently agriculture, were the essential activities which provided the necessities of food, drink, warmth and protection. The historical legacy of this utilisation of natural resources is today's agricultural and mining industries. The making of weapons, clothing, etc. gave rise to the manufacturing industry. Perhaps those people who were unable to hunt and forage survived because of their ability to make things for other or provide services and care to others. Today, manufacturing industry covers a vast range of durable and consumable goods; some are produced from raw materials, others involve further processing or assembly of components from various organisations.

Bartering your products for other people's gave rise to the industry of buying and selling for profit. The primitive marketplace has developed into the huge retail and wholesale outlets of today. Distribution of goods for profit is now a specialist industry.

Finally, as people became more sophisticated, they looked beyond their primitive everyday needs. Modern society relies on service industry to provide:

- a money interchange – banking;
- security – insurance;
- entertainment – sports, theatre, music, dancing;
- maintenance – repairs, cleaning;
- leisure – hotels, restaurants, travel, holidays, health clubs;
- culture – education, museums, art galleries.

As society has evolved, service industries have played an increasingly important part in our lives and now make a significant contribution to the economy as a whole.

A common element in all these activities is the need to survive economically, i.e. to make a profit, or at least 'balance the books' and not make a loss. Whether the organisations are profit-making or not, all are hoping to carry out their activities at a cost which can be covered by their income.

2.2 Types of enterprise

There are many different types of enterprise, each with its different characteristics, benefits and responsibilities (described later). In the United Kingdom, the terms 'private' and 'public' sector, and 'mixed economy' are used. The approximate equivalents of these categories will exist in most countries. The private sector covers businesses which are not under state-control, whilst the public sector comprises government departments,

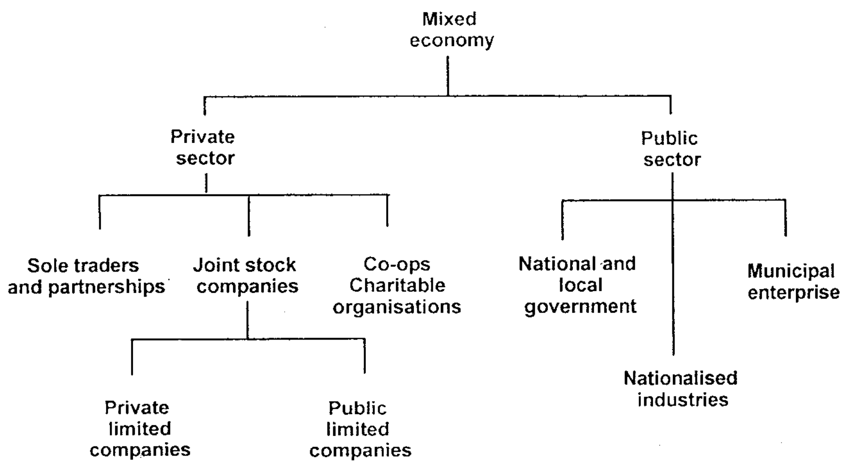


Figure 2.1 Types of organisation.

and state-run or nationalised industries. A mixed economy has both sectors represented. The private sector consists of:

- ❑ sole trader and partnerships;
- ❑ joint stock companies, which can be:
  - private limited companies (ltd)
  - public limited companies (plc);
- ❑ co-operatives;
- ❑ charitable institutions / organisations.

The public sector comprises:

- ❑ national and local government;
- ❑ nationalised industries;
- ❑ municipal enterprises (leisure centres, certain golf courses, etc. provided for public amenity).

There is a hierarchy of business units shown in Figure 2.1

2.2.1 Sole trader and partnerships

These are the smallest units in the private sector. Almost anyone can start up a business. The legal requirements are not arduous. Many businesses start as sole traders or partnerships and grow into one of the other types of organisation. Examples are farmers, accountants, builders, small shopkeepers, doctors, dentists, where large amounts of capital are not needed. The sole

	Private	Public
Size	Small to medium	Usually large
Directors	Minimum of 1	Minimum of 2
Public invitation to subscribe	Not allowed	Allowed
Minimum authorised capital	$200	$100,000

Figure 2.2 Joint stock companies.

trader is the traditional entrepreneur who provides the capital, bears the risk, controls the business and makes the profit.

A legal partnership is similar to a sole trader business. Here two or more people agree to share equally, or in fixed proportions, in the capital investment and profits/losses. Partners are not entitled to draw salaries; only to get a share of the profits. Partnerships can develop from sole traderships, often because of the need for more investment. Similarly, companies may develop from partnerships. Two main drivers of this are:

- the development of a factory system, employing many people to whom the original founders do not wish to give the status and profits of partner;
- overseas trade on a large scale, with a time lag between investment and profit, which involves greater risk and requires more invested money.

2.2.2Joint stock companies

A joint stock company is a company in which many people (shareholders) provide capital, in varying amounts and receive shares in the profits in proportion to the amounts they have invested. Limited liability (ltd.) means that the shareholders' financial responsibility to the company is limited to the fully paid-up value of the shares held. If the company is unable to meet the demands of its creditors, the shareholders may lose this, but no more. The rest of the shareholders' property cannot be taken to cover the claims of the company's creditors.

There are two types of limited joint stock company, the Private and the Public Limited Company. Their differences are shown in Figure 2.2.

2.2.3 Co-operatives

Co-operatives are membership organisations which must have a minimum number of people as members. Under United Kingdom law, the minimum is seven members; no maximum number is specified. The Co-operative Wholesale Society (CWS) is an example of a co-operative which conducts a manufacturing and wholesale business. Originally founded as a self-help group, CWS have evolved into fairly conventional retailing and wholesaling outlets but still have a membership base, from which directors are elected.

2.2.4 Charitable institutions

Charities must be legally registered, and mostly raise money from the general public. Their aims are usually of a benevolent nature and therefore attract the sympathy of potential donors.

2.2.5 National and local government

Many service functions are provided directly by national or local government. Main examples are:

- ❏ health service;
- ❏ social services;
- ❏ police;
- ❏ fire service;
- ❏ education.

Government services are paid for by taxes; local services through community charge and government subsidy.

2.2.6 Nationalised industries

Some typical nationalised industries that have been established in the UK are:

- ❏ British Broadcasting Corporation;
- ❏ British Rail;
- ❏ British Coal;
- ❏ Bank of England;
- ❏ Post Office.

All were set up as a means of providing essential services economically to the general public. In theory, they are self-financing and self-accountable, but in practice they generally report to a government minister and many require additional government finance in order to remain viable.

2.2.7 Municipal enterprise

Many local government services are, in theory, self financing, e.g. transport and entertainment. However, in practice they rely to an extent on community charge subsidy.

2.3 Business functions

Within any business there are a number of identifiable functions.(Figure 2.3) In a sole trader business, all functions may be carried out by the entrepreneur him/herself. In a large corporation, whole departments of hundreds of people may be devoted to a single function. The common functions are listed below.

- production;
- sales;
- purchase and supply;
- financial and management accounting;
- human resources (personnel);
- marketing;
- research and development;
- information systems (management services);
- corporate management.

2.3.1 Production

In many organisations, this is the purpose for which the business was established. Whether it is the manufacture of cars or the building of tiny integrated circuits, production is at the heart of the business. In service industries, the service itself is the product. An organisation usually has a wide range of products, each of which may require the use of a variety of production facilities. Production management therefore has to ensure that all resources are used as effectively as possible. This does not necessarily mean a high production of high turnover products, since some low turnover items may produce a higher profit. In many instances, very complex mathematical planning is needed to obtain the most profitable mix of work.

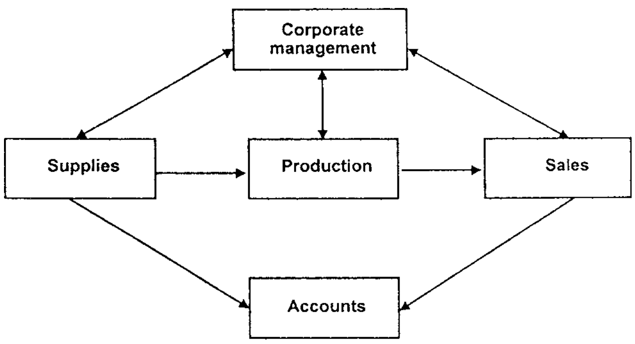


Figure 2.3 Business functions.

Often associated with production is the specialist function of production engineering, ensuring that the right machinery and tools are available to produce the products.

2.3.2 Sales

The second function essential to the organisation is sales. An organisation will survive by selling what it produces or can supply as a service. The salespeople must ensure a good inflow of customers' orders for the company's goods and/or services. They must have a close liaison with the production side of the organisation to ensure that orders taken can be met.

Distribution may be a sub-function of sales or production depending on the scale of the operation. Its prime objective is to ensure the flow of goods from production to customer. This may require a simple transportation exercise, or it may involve a complex retail distribution set-up involving many hundreds of shops, depots or other outlets. The use of the Internet for sales and distribution of products and services is now a major part of many businesses.

2.3.3 Purchase and supply

In order to make products, it is necessary to buy in some or all of the raw materials, components or services required. This involves obtaining the materials, etc. at the right price, at the right time, and in the right quantities. This area will include warehousing, stock control, and possibly the inspection and testing of goods prior to acceptance.

2.3.4 Financial and management accounting

The finance system of a business may be described using an hydraulic analogy. In Figure 2.4, the boxes represent tanks which store water (money) while circles represent drains. The boxes to the left of centre correspond to the items on the left-hand side of the balance sheet (the source of the funds). The circles on the right show where the funds have gone, as the right-hand side of the balance sheet (the application of funds).

Before a company can begin operating it must acquire capital from shareholders, or loans from banks or other institutions. Some of this capital will be used to acquire fixed assets such as buildings and equipment. The rest is required to pay the running costs of the business. The first drain is overheads, maintenance of fixed assets and servicing of loans, and staff wages/salaries. After this, money can be used to purchase materials, but this is not quite the same as the previous drains because the materials form an asset. The stock tank is being filled up. Also, it is quite common to acquire the stocks before spending the money, thus providing another source of funds – creditors. These are suppliers who expect to be paid within, typically, thirty days, but not immediately.

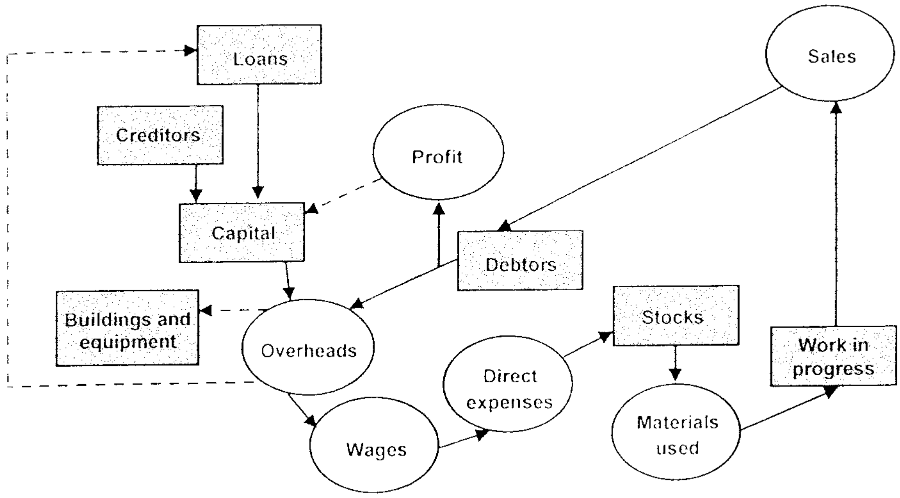


Figure 2.4 Financial accounting.

From here materials used are drained to the work-in-progress tank at the bottom of the system. Here the assets will stay unless there is a pump to lift them to a higher level. This pump is the sales effort. Once the goods are sold, the value goes into another tank labelled 'debtors' and their value is again converted into money which can re-enter the cycle to provide for overheads, etc. Hopefully, more money will come out than went in and the excess can be siphoned off in the form of profits. Some of these profits may be ploughed back into the company to form part of its capital.

Figure 2.4 demonstrates that sufficient funds must be provided at all times to keep the system primed. Once the system has been filled it should be able to operate without further filling, although it is wise to have something left in reserve tanks for emergencies.

The financial manager is concerned to see that the organisation has adequate finance and that this is properly used. He/she must provide the capital, dispose of the profit and do this in the most economical way. To provide money they may draw on shareholders for long-term capital, long-term loans from banks or other institutions, bank overdrafts or creditors. Profits may be used to repay loans, to increase the size of the business, to pay dividends or maybe to make investments outside the company. The financial manager has to ensure that capital is available to enable the business to turnover goods and money in order to make a profit.

The financial accountant is also concerned with the company finance system, but at a very much lower level of detail than the financial manager. It is their task to record and organise the details of the many thousands of transactions which take place in the course of business life. Having done this they have the means to continuously monitor the operation of the company finance system. They need to know at all times how much money is in the various tanks and how much is going down the drains. By the manipulation of 'valves' in the system and adjustments to the 'pump speed' they keep it going smoothly; for example, by reducing stock levels, chasing debtors for faster payment, blacklisting bad payers, etc.

Conventional accounting practice requires that once a year (or more often) the financial accountant should produce a complete status picture of the system. This picture is the balance sheet. Unfortunately, the system is a dynamic one and a snapshot at one point in time will not necessarily give a good representation of the general well-being of the system. Financial accounting is therefore concerned with reporting on the utilisation of capital to make and sell a range of products which cause cash to flow through the business.

A company's profit is obtained by producing goods or services, so there is a need to relate products to money. This is the function of management accounting. Ideally, it would be nice to know the exact cost of producing every item. The main costs involved are labour, machinery and materials, but there are a vast number of other costs such as heat, light, etc. Since a large number of items are being produced, it is not possible to cost everything accurately; this would be too expensive an exercise. Consequently the management accountant must use a variety of indirect methods to approximate a true relationship. One technique involves setting up a number of standard cost rates. After the goods have been produced, he/she must watch carefully for variance from these. Any major variances must be carefully investigated and, if necessary, the rates changed.

The role of the management accountant is to relate work-in-progress to profits by means of cost rates. The subject area of financial and management accounting is a complex specialist area and is not covered further in this book.

2.3.5 Human resources(personnel)

All organisations rely on people. The human resources (HR) or personnel function has developed from management's view that certain areas of people-management, such as welfare, grievances and sometimes training, could best be handled by a specialist function. The main driving force behind this in the United Kingdom is the mass of legislation relating to people at work. This includes the Offices, Shops and Railway Premises Act; Health and Safety at Work Act; Employment Protection Act; Sex Discrimination; Equal Opportunities, etc.

The HR function is concerned mainly with:

- manpower planning;
- recruitment and selection;
- education and training;
- wages and salaries;
- employment conditions;
- industrial relations;
- termination procedures.

2.3.6 Marketing

This term covers the whole area of market research, product planning, advertising, sales and public relations. It could be described as the 'front end' of any business. Marketing is the process of determining consumer demand for a product or service, motivating its purchase, and distributing it for ultimate consumption at a profit.

For a sole trader business, like a small baker's shop for example, the proprietor is in constant touch with his/her customers, and therefore their comments on the products can immediately affect the next production schedule (batch of cakes). In the larger organisation, this becomes a specialist function called market research. We have test marketing, consumer research, sales analyses, salesperson's reports, and scientific methods of collating all this data for the specific purpose of identifying the potential customer and their requirements.

Advertising seeks to motivate the sales to the customer and the retailer, using various media such as:

- direct mail;
- display advertising (posters, placards, hand-bills);
- newspapers and magazines;
- television, radio and cinema;
- exhibitions;
- sponsorship.

Public relations is a more subtle form of advertising. It endeavours to enhance the company image with the public by obtaining good media coverage and will usually be responsible for answering all customer queries and complaints.

2.3.7 Research and development

A research and development department provides a link between marketing and production. It obtains its data from market research, and in the light of customer requirements it attempts to improve the company's product. It experiments with new production methods, produces design prototypes and will advise production when one of its ideas is finally ready for a live production run. It will also experiment with ways to make a product more economically without lowering its quality.

2.3.8 Information systems/management services

A department is needed to co-ordinate all service functions provided to the departmentswith responsibility for the functions above. Such services include: Information Technology (IT), Operational Research (OR), Organisation and Methods (O & M) and Work Study (WS). In the past, production typically controlled work study and financial accounting controlled the computing facilities. As a consequence, this probably limited their effectiveness to the company as a whole.

The management services concept ensures that there is a fair sharing of these services between the different departments and functions of the organisation, in line with business objectives and priorities.

2.3.9 Corporate management

It is the responsibility of corporate management to co-ordinate all these functions. A working board of directors should have representatives from the main functions shown in Figure 2.3, to decide on priorities and resolve conflicting issues.

For example, a frequent cause of conflict is the differing requirements for effective working of the production and marketing operations. Production find it easier to have a steady volume of output throughhout the year, with the minimum number of varieties. If demand expands and they have to order new equipment, they will hope that sales can take the whole output of the new machines.

Conversely, marketing prefer a wide range of products, so that they will have a better chance of handling a 'winner'. They would rather move slowly with new lines, until they can gauge the public reaction. If sales are seasonal, they will hope that production will be able to adjust levels to suit – otherwise there will be an excessive build-up of stocks which will affect sales promotion activities and eventual profits.

These two operational areas each have to fall in line with the financial aspects of the business. It would be superfluous of them to argue about the weekly output and sales of, say tinned carrots, if tinned carrots were no longer profitable.

Profits, however, can be elusive, and a profit can quickly be turned into a deficit by losing an important customer. Hence the need for co-ordination between the operational and financial aspects of the business.

The most important function in this regard is that of management accounting. By using money as the common denominator, it enables management to look at the whole as well as the parts, and see whether planned objectives are being achieved.

Someone in the business has to take an overall view, and decide on priorities. The Information Systems (IS) or Information Technology (IT) department may think, for instance, that an immediate computer enhancement is vital, while the production manager can make an equally good case for new machinery for the factory; and there may be insufficient capital available to meet both demands. This kind of decision usually rests with the board of directors, although the responsibility for preparing and co-ordinating the plans leading up to such a decision would probably rest with management accountants, who, in their turn, would need to work very closely with the operational departments concerned.

Referring back to the hydraulic analogy, it is management's task to determine the capacity of the tanks and set the value to ensure the required rate of flow. They should also receive the feed-back information so that the system can be adjusted as circumstances change.

2.4 Objectives of business

Every business, whether it has formally established objectives or not, will be working towards some goal. The entrepreneurial may state this as:

❑ to make money;
❑ to provide an enjoyable livelihood.

Limited companies have an obligation to provide a livelihood for their shareholders. Most would feel a similar obligation to their employees (who are also protected by government legislation). In the public sector the objectives identify more with the services being provided, with perhaps an underlying objective to remain profitable, or within budget, where possible.

In other organisations the emphasis is on innovation: identifying new markets and creating new products. In every sphere there has to be a 'moving with the times', so although the main objectives remain static, the subsidiary objectives may change.

Systems will also change to meet the changing objectives; influences such as new legislation, trade fluctuations, company mergers and market developments. The objectives involve three sets of people, the customers, the employees, and the shareholders (or in the case of the public sector, the financially responsible body).

Other ethical considerations may be part of the constitution or 'terms of incorporation', established when the organisation was first set up. The company should consider whether it would, for example, stoop to any means to improve its market share against competitors. For example, would it indulge in industrial espionage, sabotage, etc? Would it produce any kind of product

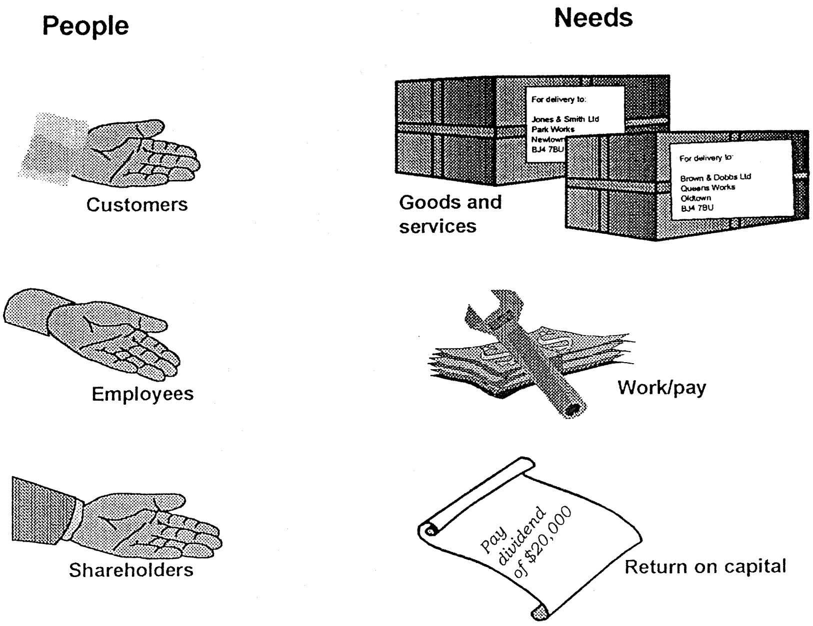


Figure 2.5 People in business.

(addictive, ponographic, harmful to health) to remain financially viable? Many of these considerations are often left unspoken and unwritten, but most reputable organisations would probably be very much influenced by such considerations.

2.5 Principles of business organisation

The principle of organisation is merely to provide an efficient framework for undertaking functions within a business to enable the achievement of the desired objectives. (Figure 2.5).

In a very small business all functions will be carried out by one or two people. For example, in a small baker's shop (baking the bread), advertising (writing out display cards), selling (over the counter), purchasing (flour and yeast), personnel (taking on new staff, paying wages) and perhaps even accounts ('doing the books') – although even very small organisations would probably employ an accountant to give specialist advice on taxation, etc.

As the business becomes larger, greater specialisation creeps in. The baker might employ a shop manager to look after selling, in order to concentrate on production. If further branches are opened then the baker may adopt a more strategic management role, perhaps maintaining a central accounting and purchasing function.

The classic theory of organisation structures puts the emphasis on specialisation without losing overall co-ordination and integration. Its ten principles, as expounded by **Lyndall Urwick** in the 1920s, still have relevance today. They are:

The principle of the objective Every organisation and every part of the organisation must be an expression of the purpose of the undertaking concerned or it is meaningless and therefore redundant.

The principle of specialisation The activities of every member of any organised group should be confined, as far as is possible, to the performance of a single function.

The principle of co-ordination The purpose of organising, as distinguished from the purpose of undertaking, is to facilitate co-ordination, unity of effort.

The principle of authority In every organised group the supreme authority must rest somewhere. There should be a clear line of authority from the supreme authority to every individual in the group.

The principle of responsibility The responsibility of the superior for the acts of his/her subordinates is absolute.

The principle of definition The content of each position, including the duties involved, the authority and responsibility contemplated and the relationships with other positions, should be clearly defined in writing and published to all concerned.

The principle of correspondence In every position the responsibility and authority should correspond.

The principle of span of control No person should supervise more than five direct subordinates whose work interlocks.

The principle of balance It is essential that the various units of an organisation are kept in balance.

The principle of continuity Reorganisation is a continuous process; in every organisation specific provision should be made for it.

2.6 Organisation structure

Various organisational relationships can be defined in this formal structure.

In the *line relationship*, each person below the chief executive is accountable to someone else. Each supervisor has a line relationship (or command relationship) with their superior and their subordinates, thus providing a chain of command throughout the organisation. The line relationship gives the official line of communication from the top to the bottom in the organisation, the line of accountability and the line of responsibility.

The *functional relationship* (or 'staff' relationship) is usually adopted where business efficiency can best be achieved by transferring management responsibility for one person to more than one manager, according to assigned specialisation. Authority still flows from the top, but the supervisory staff, organised functionally, have a degree of authority over all personnel at the level below. The aim is to promote co-operation in order to achieve greater efficiency.

A specialist department may carry out various tasks which do not necessarily involve the exercise of authority, especially in the provision of advice or of a service; this type of relationship is called a *specialist relationship*. However, if the advice is always taken, or the service always used by management, the specialist department is exercising authority and therefore falls into the category of a functional relationship.

Finally there is the *lateral relationship*. The classic approach acknowledges that relationships exist between people at roughly the same level in the hierarchy to sort out problems and co-ordinate efforts. **Urwick** comments:

> 'It is both right and proper that every organisation should have its formal scaler chain, just as every well built house has a drainage system. But it is as unnecessary to use the formal channels exclusively or primarily as the sole means of communication as it is unnecessary to pass one's time in the drains.'

Such relationships are not normally shown on organisation charts.

All these relationships will exist in large organisations since the range of skills required is too diverse to be found in individual line managers. Specialist staff are required to define strategy, propose solutions, prepare plans and initiate change for line management.

2.6.1 Departmentalisation

The choice of approach to departmentalisation rests on the type of activity performed, the objectives of the company, the relationships of that particular department to the overall goals, and external factors such as the availability of labour, land or materials, etc. There are five main approaches to departmentalisation:

Function This divides the company into its basic functions – finance, sales or marketing and production – and is mainly found at the high levels of the company. This method is often used because most companies are naturally divided into three areas. However, an element of friction or competition may arise between the functions (e.g. sales and production) as each function may try to achieve its own objectives before those of the company as a whole. This may decrease or distort the information flows between them.

Location This divides the company into geographical areas and is used when activities cannot be grouped by other methods because of their dispersion, e.g. sales territories, hotel chains, etc. This method is chosen because it is more economical, e.g. it may be cheaper to distribute components rather than finished goods, then make up the product nearer the market where there is knowledge of specific local conditions and requirements. A company may also decide to move one of its factories for other reasons, such as government grants, cheaper labour or land, etc. This obviously leads to a lack of communication as the parts of the company are physically separated from each other.

Customer This divides the company (often only the sales division) into separate units to meet the requirements of specific customers, e.g. industrial buyers or private individuals. Customer satisfaction is increased in this way because of the specialist knowledge of their requirements, which may increase the number and value of orders. Again, this split can lead to competition between the areas, resulting in friction and a reduction in effective communication.

Product This divides the company (often only the production department) into departments around product lines, e.g. a motor manufacturer may have one division per model. Again, competition can lead to friction and a lack of overall co-ordination.

Process This divides the company (again often only production) into departments for each process, e.g. round particular groups of machine. This makes overall co-ordination difficult, as it is difficult to keep track of where an order has reached in the sequence of processes.

It can therefore be seen that by adopting one or several of these methods of departmentalisation, the flow of effective communication can be reduced, but the overall company may be more effective as each section contains specialists who are perhaps more able to meet the company objectives for their particular section.

2.6.2 Human Relations School

The Human Relations School of organisable theorists is critical of the classic

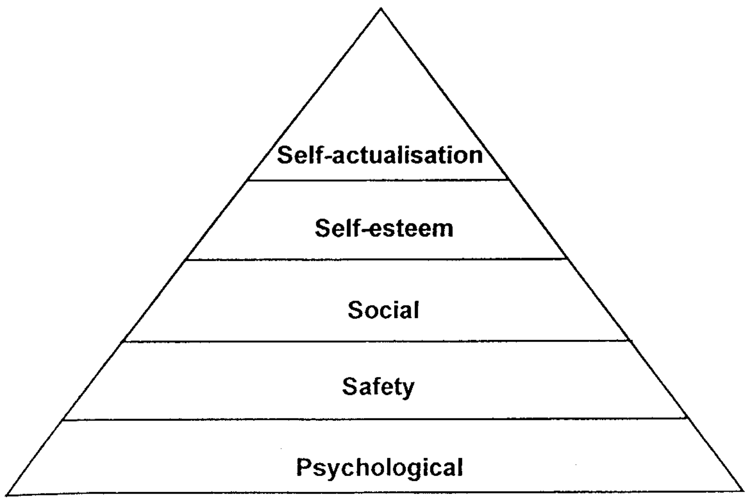


Figure 2.6 Maslow's needs hierarchy.

approach because it fails to take account of the people who are being 'organised', their behaviour and their needs. It is likely, for example, that people will react differently to different organisation structures. The classic view tends to see the humans in the organisation as willing followers of rules, whereas in practice people often pursue their own interests and ideals and build up a social structure to enable them to achieve them. The classic approach tends to concentrate on formal organisation structure and to ignore how the structure operates in practice.

There are a large number of such theorists; some writers and some of their contributions are discussed now.

Individual motivation Many of the theorists have concentrated on individual motivation and attempted to identify the factors which predispose individuals to behave in certain ways. **Abraham Maslow's** theory of motivation is based on a hierarchy of man's needs. At the lowest level there are basic physiological needs (e.g. for food, shelter, sleep); at the next level come safety needs (e.g. for security, health); above those come social needs (e.g. for love, affection); next there is the need for self-esteem (e.g. for prestige and self-respect); and finally there is need for self-actualisation (e.g. for achievement and personal development). **Maslow** proposes that the individual satisfies their needs in a priority sequence by starting at the lowest level and working up through the hierarchy. The main implication of the theory is that work must be organised to allow individuals to achieve their higher psychological needs, as shown in Figure2.6.

Frederick Hertzberg followed up **Maslow's** ideas by conducting extensive surveys of job satisfaction. As a result of these he propounded a theory of motivation which identified certain factors which must be present to avoid job dissatisfaction (called hygiene factors), and other factors which must be present to give job satisfaction (motivators). The hygiene factors are mainly related to the context of the job (e.g. salary, status, security, working conditions) and the motivating factors are mainly related to the content (for example, responsibility, achievement, self-growth and challenge). **Hertzberg** argues that the organisation of work should concentrate on improving job content because this will increase the employee's satisfaction (whereas improving job content only removes dissatisfaction). **Hertzberg's** ideas have been developed into a 'Theory of Job Enrichment', which has been widely applied.

Douglas McGregor developed a view of motivation which he called Theory X and Theory Y. The Theory X view of man is that he is lazy, selfish, stupid, and responsibility-avoiding; this suggests that a carrot and stick approach to motivation is required. The Theory Y view of man is that he is ambitious, keen, capable and responsibility-seeking; this suggests that he is motivated by self-control rather than external controls. Though Theory X and Theory Y represent opposite extremes of a spectrum and, of course, each individual is to be found at a different position on the spectrum, the essence of McGregor's ideas is to point out strongly that preconceptions about human beings can strongly influence their performance, and that perhaps a willingness to move towards the Theory Y view might improve organisational effectiveness.

The ideas of **Maslow**, **Hertzberg** and **McGregor** have not been received without criticism – and in some cases total rejection – but they have stimulated a wider view of management's approach to organisation.

Work groups Other behavioural scientists have concentrated their research on behaviour in groups, on the grounds that people in organisations do not operate as individuals in isolation, they come together in formally-organised or informal groups. The most influential ideas in this area emanate from the so-called Hawthorne Studies carried out in the 1920s and 1930s in the Hawthorne Works of the Western Electric Company (USA), (**Urwick** & **Brech**). The main emphasis of these studies were to draw attention to:

❑ the role of work group norms in governing worker performance;
❑ the pressure on individuals to adjust to and accept the group norms;
❑ the influence of informal leaders on setting and enforcing group norms.

Again, the Hawthorne findings have been subjected to much criticism.

Leadership A third area to which the Human Relations School has drawn attention is the effectiveness of leadership in organisations. Most of the work in this area has concentrated on:

- ❑ the extent to which a manager is employee-oriented or production-oriented, and the implications for improving work performance;
- ❑ the degree of influence that a subordinate has on the decisions of his/her supervisor (i.e. democratic vs autocratic management). The ideas of **R Likert**, **F Fiedler**, *et al* have been most influential in this area, but the main impact has been on management training programmes rather than changed organisation structures.

Organisation structure The major influences of the Human Relations School on organisation structure have been in four areas:

- ❑ flat *versus* tall organisation structures;
- ❑ reduced specialisation;
- ❑ increased worker participation;
- ❑ recognition of informal structure.

Tall organisational structures were the result of the classic approach and its emphasis on a limited span of control. They require more vertical communication, because each supervisor has fewer subordinates, and so they were criticised by the Human Relations School for removing the top management too far from the action. Flat structures have been recommended to overcome these problems and to take advantage of less formal relationships. The problems with flat structures are that they require much better co-ordination and control, and they may lead to overload (especially at the top). Flat structures seem most appropriate to small organisations or those which are working in an environment of considerable uncertainty.

The ideas of the motivational theorists have encouraged organisations to move away from specialist departments towards more wide ranging and integrated departments. These give the individual increased opportunity for growth, challenge and responsibility.

Worker participation is another development encouraged by the Human Relations School – although its effectiveness is often doubted and it is not always welcomed by the workers themselves. The arguments used to support its introduction include increased productivity, reduced conflict, and because of higher educational levels workers feel capable of making a contribution to decision making and therefore demand this role. However, opponents of participation argue that it does not always increase commitment or productivity, but it does tend to increase the time for decision making and to blur responsibility for decisions. Industrial democracy is a phenomenon which will grow and as such, it will have a significant impact on organisational structures.

The relationship between the formal and the informal organisation has been explored considerably by the Human Relations School. **Amitai Etzioni** has defined the two types. Formal organisation is:

> '. . . the pattern of division of tasks and power among organisational positions and the rules expected to guide the behaviour of participants by management . . . it is the blueprint according to which organisations are to be constructed and to which they ought to adhere.'

He sees the formal organisation as the explicit definition of tasks and roles and their co-ordination and control in the interest of achieving certain goals; it can be considered to exist independently of the particular people who are members of the organisation.

Informal organisation is:

> '. . . either the social relations which develop between staff and workers beyond the formal ones determined by the organisation . . . or the actual organisational relations as they evolve as a consequence of the interaction between the participants.'

This describes the informal organisation as being not planned or rational; it develops as a result of the social needs and values of its members. It exists partly to give individuals (and groups) a means of exerting control over their work situation, and partly because the formal organisation does not cover all organisational activities.

2.6.3 Contingency theorists

The most recent contributions to the study of organisations have come from the contingency theorists who suggest that organisation structures should reflect current environmental circumstances. The main factors they consider to affect organisations are technology, innovation and uncertainty.

Studies of the relationship between technology and structure of organisations have been pioneered by **Trist** & **Woodward**. **Trist** has developed the idea of a socio-technical system in which technology constrains methods of organisation of work, but these are in turn constrained by what is socially acceptable and economically feasible. Hence, technical, social and economic factors need to be taken into account in changing organisation structures. **T Burns** and **G M Stalker** looked at a number of firms in the electronics industry and measured the success of their organisation structure in coping with changing environments. They concluded that an organismic structure coped more readily with innovation than a mechanistic one. The different approaches to coping with the organisational environment are shown in Figure 2.7.

The contingency theorists as a whole are drawing attention to the complexity of organisations and arguing that no single approach (whether it be classical or human relations) adequately meets the needs of all organisations.

Since organisations are at present facing a time of rapid change, steps are being made towards more organismic, participative structures.

Mechanistic	Factor for comparison	Organismic
Specialist	Structure	Generalist
Defined roles	Tasks	Fluid roles
Structured, vertical and rigid	Communication	Flexible, lateral and transient
Formal and rare	Meetings	Informal and frequent
Written	Medium of communication	Spoken
Directive	Decision making	Participant

Organisational environment coped with

Static ◄——————► **Dynamic**

Figure 2.7 Mechanistic and organismic organisation structures.

2.6.4 Organisation structure and the systems analyst

The systems analyst needs to be aware of the organisation structure and its implications in order to be able to design effective information systems. In particular they need to understand the problems that hierarchical structures can create for information flow, and to appreciate the importance of informal information flows.

Hierarchical structures One problem of hierarchical structures is that information flows are usually determined by line relationships, and therefore it is very difficult for consistent and accurate information to reach the top of the hierarchy. Different departments adopt different approaches to problems even though they are making use of the same information. Thus reports which are summarised from a number of departments as they move up the hierarchy may be based on different timescales, different interpretation of rules, different coding systems, etc.

The boundaries of departments stand to become jealously guarded and lead to duplication of effort and information. For example, lack of trust (or an inappropriate structure) may cause the sales, accounts and production departments each to maintain details of customers' orders (often with quite different coding systems).

It is difficult to produce *ad hoc* reports (to meet a sudden specific enquiry from top management) across departmental boundaries.

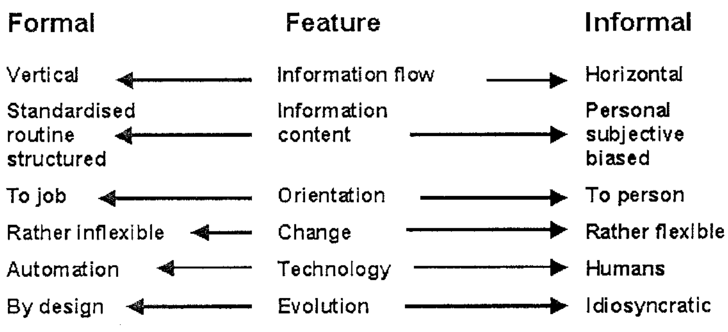


Figure 2.8 Formal and informal information.

Hierarchical structures tend to encourage very formal, highly structured information flows in which information is filtered (possibly in the wrong way), inconsistent and incomplete.

Another problem encountered in formal organisation structures is the handling of boundary tasks. An example is the collection of overdue accounts from customers; the responsibility for this activity must be clearly allocated. The sales ledger department may feel that it has enough to do without writing reminders to customers, especially at year end, and may feel that this is the job of the salespeople. They in turn will not wish to upset customers by aggressively chasing up bad debts because future orders and commission may be jeopardised. Each department believes the other to be responsible for bad debts, and the organisation as a whole suffers.

The final problem of hierarchical structure is *empire building*. The frequent result of specialisation and departmentalisation is easier control of the individual departments at the expense of co-ordination and corporate control. The whole organisation becomes difficult to manage because departments are self-interested, jealous and blinkered, and their objectives slide. As a result, information tends to reflect departmental objectives and timescales rather than those of the company.

2.6.5 Formal and informal information

Systems analysts spend most of their time dealing with formal information flows and so they need to realise the limitations of these.(Figure 2.8) Formal information typically flows along the lines of the organisation chart vertically. It is usually standardised in a rational way and structured to meet job needs, it is often inflexible to short-term changes, it makes use of mechanical aids, and is usually specified by designers.

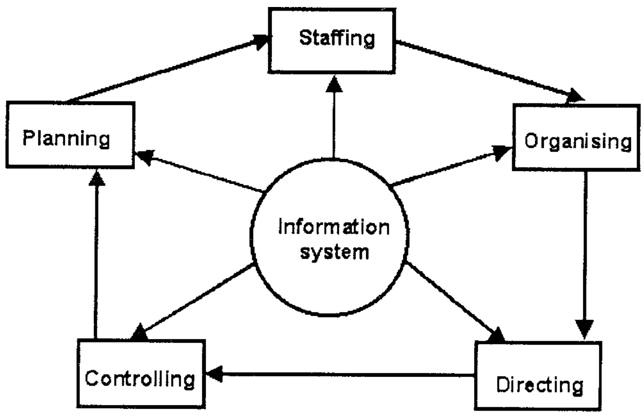


Figure 2.9 The management process.

Informal information, however, is personal and directed to the recipient as a person rather than a job incumbent. It is not standardised or rational but reflects subjective judgements and perspectives. It is communicated on an *ad hoc* basis and so is more flexible. It is not capable of mechanisation, and it grows idiosyncratically as part of the user's dealings with other individuals (i.e. it is not designed). Informal information is essential to fill in the gaps of formal information and to aid social interaction.

2.7 Management

In order to identify appropriate information flows in the organisation, it is necessary to analyse the tasks of management.

2.7.1 The management process

The classic view of management includes five stages:

❑ planning;
❑ staffing;
❑ organising;
❑ directing;
❑ controlling (Figure 2.9).

The planning stage is concerned with the identification and selection of objectives for the organisation, perceiving opportunities for change, diagnosing and determining procedures to achieve objectives.

- ❏ planning must be carried out at all management levels, but with greater specialisation and over a shorter timescale at the lower levels of the hierarchy;
- ❏ the staffing stage involves the definition of human resource requirements, and subsequently selecting, training and remuneration of staff;
- ❏ organising work requires the grouping of activities to be performed and structuring the organisation to carry out the activities;
- ❏ directing is nowadays thought of as people-management, i.e. co-ordinating, motivating and guiding staff;
- ❏ the final stage of controlling involves measurement of performance, highlighting of variances, observation of trends and making adjustments.

Three observations need to be made about this view of management:

1) The apparent chronological relationship between the stages is unreal, because usually all tasks are performed concurrently (for example, planning and review of plans) though perhaps on different projects.
2) The whole process requires information; in particular, plans cannot be developed without detailed knowledge of objectives, resources, environment etc., and control against plans requires detailed feedback on performance.
3) The common feature of all stages could be defined as decision making.

2.7.2 Levels of management

The number of levels of management varies in different theorists' approaches but usually there are three levels – top, middle and junior (or supervisory), as shown in Figure 2.10. Top management consists of the chief executive and the heads of functions – whose job is to interpret company objectives and to manage the organisation to achieve its objectives. Middle management comprises departmental heads and assistants to top management (especially specialist functions) whose task is to achieve co-ordination and to control performance against the overall objectives. Supervisory management includes supervisors, section heads and foremen whose rôle is to allocate work to individual employees or groups of employees and to ensure that it is carried out. All levels are involved in a certain amount of planning but this is greater at the top level. Man-management tends to have greater importance at lower levels of management.

2.7.3 Planning and controlling

A plan can be defined as 'a pre-determined course of action'. This starts with an awareness of an opportunity, is then expressed as a set of objectives, which in turn leads to the identification of alternatives These alternatives are

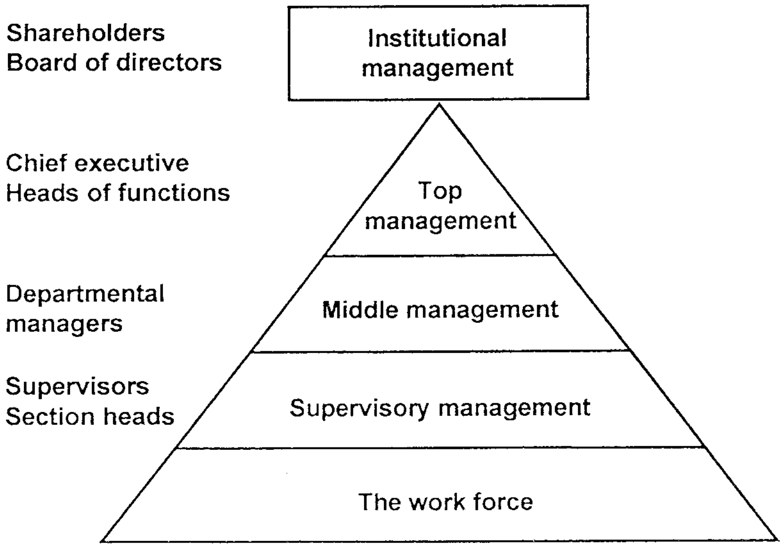


Figure 2.10 Levels of management.

evaluated and one is selected for implementation. The success of the plan must be measured and its implementation adjusted in the light of changed circumstances, inadequate performance, etc. Planning and controlling can therefore be viewed as integrated parts of the same process. This is shown in systems terms in Figure 2.11.

Planning can be analysed from a number of viewpoints in terms of its requirement for information. The most common viewpoint is timescale, and usually planning is seen as long-range (strategic), medium- range (tactical) or short-range (operational).

Strategic planning is carried out by top management and usually examines organisational developments five to ten years in the future, e.g. methods of growth, new technologies, new products, mergers, etc.

Tactical planning is carried out by functional managers and attempts to determine policies for achieving the strategic plans. For example, it might look at plans for introducing a specific new product over a three-to-five year period.

Short-range planning involves operational managers (department heads) in deciding on procedures to achieve objectives over the next year. For example, short-range planning may well be concerned with inventory levels or production levels.

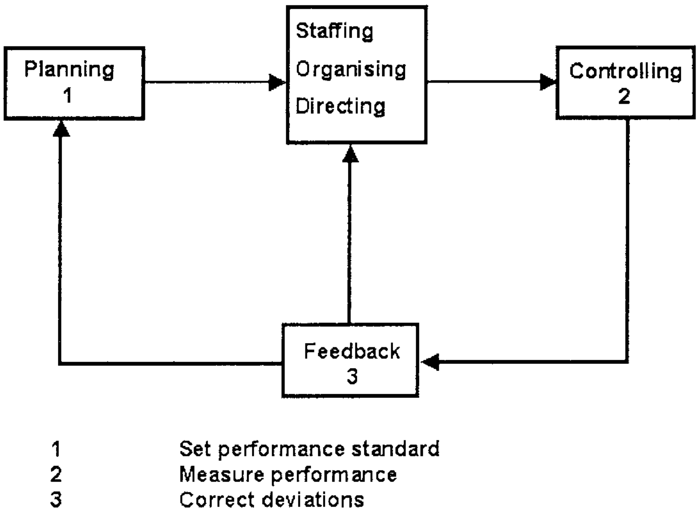


Figure 2.11 Planning and control

The planning process for the organisation can be seen as a structured process as in Figure 2.12.

Surveys of the current position of the organisation, the external environment and the corporate objectives lead, by analysis and debate, to a long-term corporate plan which is analysed into a medium-term operational business plan. This is reduced to a set of short-term operational plans usually expressed as budgets. The information needs of these various stages (at each level) consist of knowledge of current and past performance, forecasts of future performance, views of government policies, technological developments, market changes, and a feel for the political, social and economic climate.

Control of performance against the plan involves three main stages:

- ❑ setting of performance standards (usually expressed as financial budgets, sales targets, timescales or physical measures);
- ❑ measurement of performance (usually by observation, interview or analysis of reports from subordinates);
- ❑ correction of deviations from the plan (usually by changing plans, standards of performance or by revising staff/resource allocations).

Clearly, this requires detailed information on performance at the lowest level of the organisation.

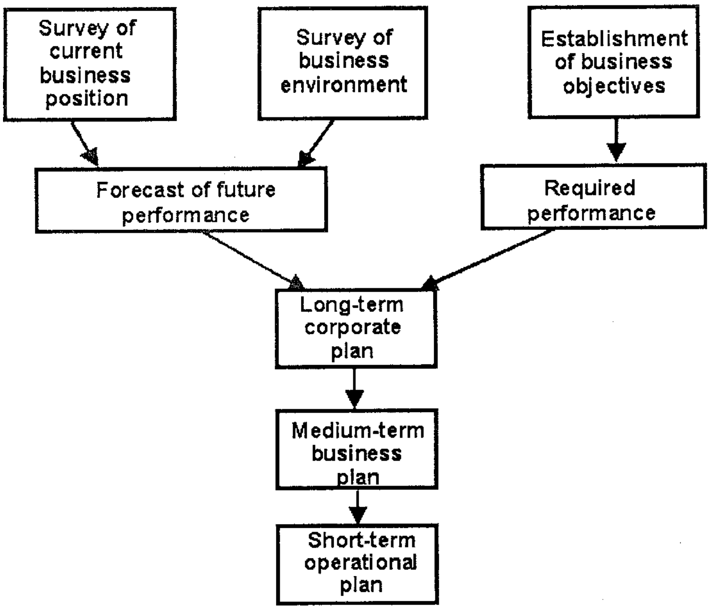


Figure 2.12 The Planning Process.

2.7.4 Decision making

In many ways decision making is the same as planning, since planning involves the taking of several decisions and each decision involves an element of planning. A definition of the word 'decision' will help to illustrate this point:

> A course of action chosen consciously from available alternatives for the purpose of achieving a desired result.

The quality of managers' decision making will depend very largely on the quality of information with which they are supplied.

2.8 Information within business

As outlined earlier, every organisation consists of a number of business functions: sales, production, accounting and so on. In order to co-ordinate these functions effectively, information must be passed between them. Information is needed to enable a business to produce and sell its products, to assist decision-making and to enable controls to be effected, both at the functional level and overall. Information itself is a corporate asset. Since the

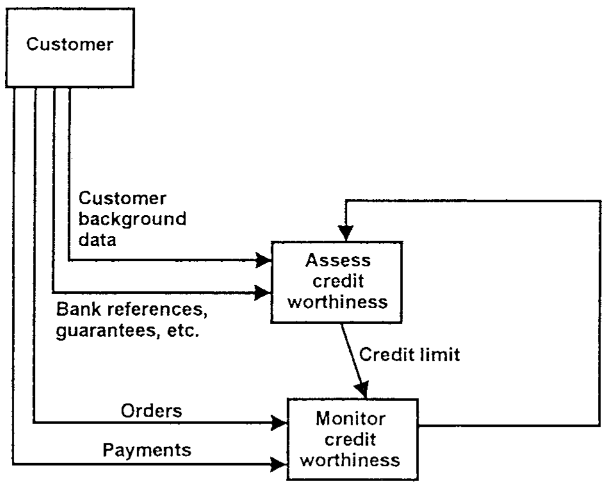


Figure 2.13 Credit control.

advent of computers, the major enabler of information flow at all levels is the computer department, often referred to as Information Systems (IS) or Information Technology (IT) departments.

The function of the IT department, is to collect and store 'data' the business requires, but to provide 'information'.

2.9 What is data, what is information?

Data is a set of characters/figures without any particular coherence. Information is those same characters or figures in context. The figures 01072001 are data. When it is evident that they represent the date your manager is due to retire from work, they become information.

One of the purposes of systems analysis is to identify the data which the new system will require. There may be a known problem within the organisation; the system could provide certain outputs of information to help solve that problem. What data is needed to provide that information, and where can it be found?

Take credit control as an example: the credit controller needs to have customer information. In the first instance, this will be bank references and other background data relating to a customer so that the controller can assess the customer's credit worthiness. Then, as a continuing process, they check data on customer orders and payments so as to continuously monitor their credit worthiness, and if necessary, readjust their credit status with the organisation.

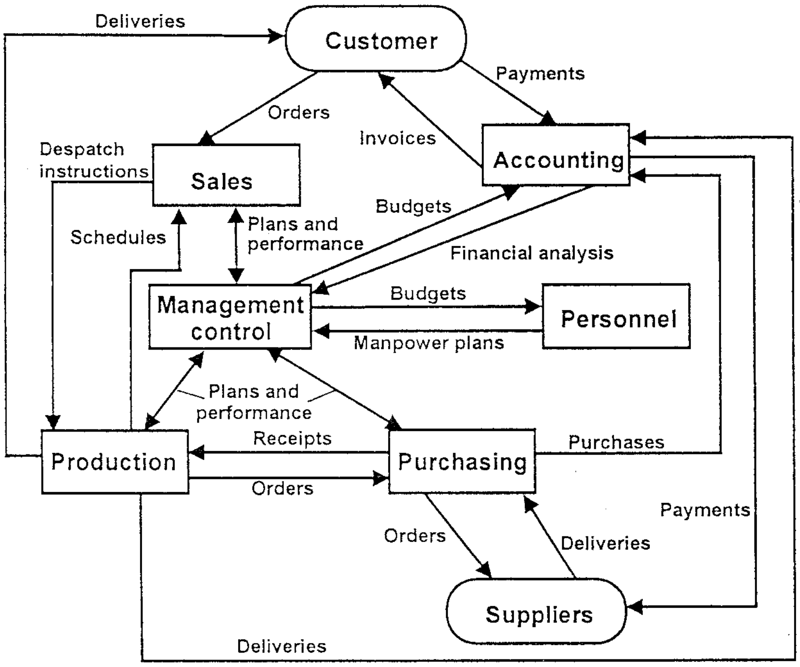


Figure 2.14 Simple model of a business.

The credit controller formulates their information requirement, the analyst considers how best that information could be produced from the data available, and between them they should discuss how it might be processed. What rules can be formulated to make the system function? Should an order be stopped automatically if a customer reaches their credit limit? Should orders be automatically released if a payment brings them within the limit again? How are credit limits to be set? Can they be adjusted automatically, say, to equal one month's turnover? How does the controller want the information presented? A full payment history for everyone, or only when someone reaches their credit limit, or perhaps only when debts go over a certain age?

2.10 Process

Figure 2.14 is a very much simplified diagram of a business which gives some idea of how data might flow in a complete business system. In the total business context, information at one level is data at another level. To pursue the credit control example, the credit controller is going to be interested in individual debts. At a higher level however, the financial director will probably

Account	Current	>1 Month	>2 Months	>3 Months	>4 Months
Adamson	3,421.55	2,787.72	416.50	45.45	-
Brown	2,461.25	1,256.40	933.33	750.20	450.50
Carter	4,493.58	1,240.35	750.75	-	-
Davies	2,164.72	-	-	-	-
Evans	3,131.20	-	-	-	25.20
Fraser	1,326.82	2,165.40	-	-	-
Gibbs	2,444.70	-	-	-	-
Hansen	4,270.10	1,216.75	55.80	-	-
Total	23,713.92	8,666.62	2,156.38	795.65	475.70

Figure 2.15 Aged debt analysis.

want to know how the total debt problem is going to affect the company's cash flow position, and therefore a summary of all outstanding debts will be more relevant to him.

In the example (Figure 2.15), the credit controller would certainly be interested in Brown's account, possibly Evan's (an outstanding credit missing?), whereas the financial director would probably only need a summary of the bottom line.

Normally, items of data have to be related to one another to provide information for the next level up in the organisation structure. Thus the major information flows in the organisation follow the line structure; instructions pass down from management and information flows up to management. **RWRevans** conducted an interesting study of the flow of information up and down the hierarchy. His study suggests that instructions (decisions) are slow in getting down the hierarchy but do usually arrive fairly accurately, whereas most people at the bottom of the hierarchy feel that information flow up from the bottom is poor (because of filtering, etc.) and that top management does not receive a correct picture of the events/needs at the bottom of the hierarchy.

The other information flows in the organisation are horizontal and external. Horizontal information flows between people at similar levels in the hierarchy, for example the sales order office will check credit with the sales ledger department and stock levels with the stock records office. Often horizontal information flows are informal and include interaction such as rumour and gossip.

External information flows from outside the organisation. It can consist of transactions such as customers' orders, suppliers' invoices, information about any environmental activity, for example government legislation, or new products introduced by competitors. Again, external information flows can also be informal. Figure 2.16 provides a summary of information sources and flows.

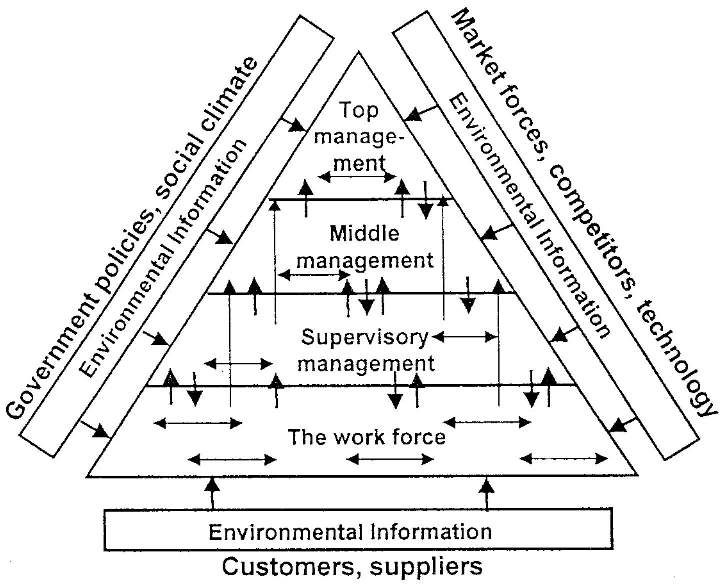


Figure 2.16 Information flows.

At the lowest level of the organisation, staff handle operational transactions such as orders, time sheets, delivery notes, sickness notes, production work-sheets, inspection advices, stock records, etc. These documents are the working papers of operational systems. For example, to obtain payment from customers for their purchases an invoice is needed, to pay an employee for hours worked a time sheet is required, or to re-order goods for the warehouse, a requisition is used.

These operational documents are used to priovide summaries of operational activity to the next level in the hierarchy. The summary information is then compared and analysed to provide management reports at the next level and so on. Thus the payroll clerks handle time sheets to produce the payroll, the time sheets are summarised to provide details of time spent on different activities in different departments for the departmental managers, and the time analyses are related to costs and production achieved for the production manager. Therefore, the information flowing up the organisation is based on the events at the lowest levels. At higher levels in the hierarchy, increasing importance is placed on horizontal and external information, until at the top management level these are at least as important as internal management reports.

It is apparent that data can be gathered from any source which impinges on the organisation's activities. It is the responsibility of the analyst to devise systems which will process that data in order to meet management's information requirements.

2.11 Purpose

What is management's rôle in relation to its information requirements? To what use will the information be put? What is the purpose of the information?

At each stage of the management process four types of information are required:

- ❏ internal company performance (sales, productions, personnel, stock turnover, cash flow, etc.);
- ❏ internal plans (sales forecasts, budgets, etc.);
- ❏ environmental (political scene, social climate, economic trends, technological developments);
- ❏ competitive (demand for the product or service, competitors' performance and plans).

The latter is also part of the environmental information.

Internal performance information consists largely of factual route reports on products made, sales achieved, money spent, employees recruited, wages paid, etc. This forms the feedback information for measuring against the plan/budgets. Internal plan information includes policy statements, forecasts, operational budgets, short term plans, etc. This gives the objectives which the organisation is setting out to achieve and against which performance is measured. The environmental or external information is more relevant at the top of the organisation and involves, for example, political trends, government legislation, social climate, technological developments, demand for products, competition, etc. This information is useful for identifying opportunities and adjusting plans.

2.12 Level

Differences can be identified between these kinds of information at the various levels of the organisation (see Figure 2.17).

At top management level, information tends to be ill structured, *ad hoc*, informal, external, uncertain and concerned with the future. At middle management level, information is more structured, formal, internal, regular, certain and concerned with the near future. At supervisory management level, information is repetitive, programmable, largely internal and has a very short time horizon.

Therefore the purpose of information changes according to the level at which it is aimed.

Paradoxically, the higher up the organisation a manager is, the less they

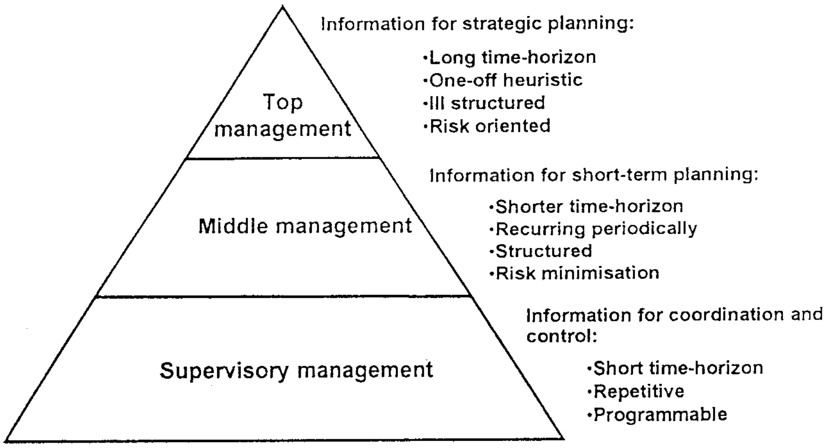


Figure 2.17 Information flows.

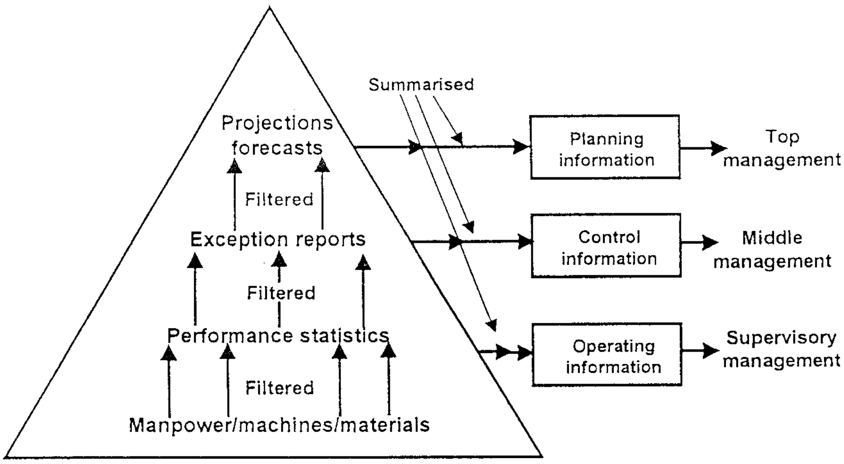


Figure 2.18 Data and information.

rely on formal information, and yet still need to be informed (at summary level) of all the organisation's activities. On the contrary, the lower the manager is in the hierarchy, the more they need to rely on precise and up-to-date information for their activities; yet they only need a very limited set of the total information for their own particular activity. Figure 2.18 summarises the level of information and the purpose to which it is put.

Supervisory management requires day to day personnel performance information, machine performance and information material usage.

At the middle management level this can be combined and summarised into exception reports, personnel, equipment, or materials above or below accepted levels, and comparisons against budget.

At the top management level, middle management information can be combined and summarised, and also perhaps at this stage projections or forecasts produced.

2.13 Content

Data and information occur in a variety of forms. At the lowest level there is the day to day information which feeds the operation of the business. This is the information required to make products or produce invoices, etc. It is detailed and specific. Most commercial IT systems deal with this type of data.

Having captured the data, it is processed to provide the types of information management requires. Typically, this information takes the form of a report.

Three types of report can be identified: the historical report, the exception report and the forecast.

The historical report might be a complete printout of statistics going back over some period in time. Often it is produced by default because the manager has been unable to define their real requirements and seeks security in the volume. 'If we have everything printed out we won't miss anything' or 'I don't know exactly what I'm looking for until I've seen it'. Alternatively, it can be a comparison exercise; this period compared to the same period last year, or perhaps an exercise in information retrieval, 'How many XYZs were exported last February?'. The on-line system, with access to a corporate database, has fortunately made the historical listing largely unnecessary, and an exception report is much more likely to be easily obtainable. When it comes to recording historical data, decisions may have to be made on how long full details need to be held. Microfilm, microfiche and compact disc give the possibility of off-line storage of archive material, with only historical summaries being held on-line. Now, with the advent of optical discs, historical data can be stored in vast quantities and still be easily accessible.

The exception report is a powerful tool. Provided that the manager is able to define their requirements, it will save them valuable time. The computer sorts and selects only those items of information required, saving the manager or a clerk much effort. The computer system can be programmed to ignore normal working and just pick out those events which are unusual or erroneous. Rules have to be defined to establish what constitutes an exception. Exception reporting is particularly useful at the middle management level, in such areas as production, purchasing and sales.

The forecast uses current and historical data to extrapolate and produce a

January	9	December	7
February	10	January	8
March	12	February	11
April	13	March	10
May	11	April	15
June	15	May	9
July	11	June	16
August	10	July	10
September	8	August	7
October	9	September	4
November	5		

Figure 2.19 Monthly sales.

January	1	December	3
February	0	January	2
March	2	February	1
April	3	March	0
May	1	April	5
June	5	May	1
July	1	June	6
August	0	July	0
September	2	August	3
October	1	September	6
November	5		

Figure 2.20 Sales variation.

projection of future activity. Many of the developments in forecasting systems have come from the area of stock control, where it is very important to estimate future item sales so that enough stock is held to cover the projected demand. In stock control, how much stock should be available at all times? Obviously enough to last until a new supply arrives. If there is no merchandise on order (that is, still to be delivered) enough stock is required to last for one 'lead time' (that is, to cover the period between making an order and receiving the delivery). In order to find out what the desired stock level is, it is necessary to make an estimate of future demand. For example, if the lead time for a given article is two months, stock of at least the forecast sales for two months ahead is needed. This forecast may be more or less accurate, and allowance must be made for this fact. Figure 2.19 shows the past sales of an article, month by month.

It can be seen that the average monthly sales are 10. The variation on this average in the past is shown in Figure 2.20.

There is an average error of 48/21, i.e. 2.3 per month, so the forecast for this item might be expressed as 10 ±2.3.

Consequently, a certain amount of extra *buffer* stock is needed to cover probable forecast error. There are many packages available containing complex formulae, such that this area warrants a study to itself.

2.14 Validity

Data, and hence information, is of little value unless it is accurate, timely and complete. Controls need to be established in order to ensure the accuracy and validity of input, output and data stores. In on-line systems, there is a great advantage in being able to update data immediately when a change is known. A purchase requisition, for instance, can be based on immediate knowledge of sales and goods received data, if these systems are linked to the computer files. Previously there was an inherent delay in updating files; and consequently many decisions were based on data that was out of date. However, the desirability of control is still important. Checks for accuracy and validity need to be made on every item input and because updating is immediate, it is particularly important to have a means of correcting errors when they are found. For this reason there must be adequate security built into the system to allow the files to be restored when they are corrupted by erroneous data.

Another problem of security in relation to information is that of file access. Only those people who are authorised to read and/or update files must have access to them. Having access checks, such as passwords or badge readers, means that the danger of files being accidentally corrupted is reduced, as is the threat of vital information falling into the wrong hands.

Equally important to an information system is the consistency of data. When data is compared, it must be 'like' data, it is no good comparing 'unlike' data and expecting to obtain useful results. To help the analyst understand the nature of the data required by a system, the techniques of Entity Analysis and Relational Data Analysis have been developed (see Book 2, Systems Analysis Techniques). Using these techniques, the analyst derives the component parts of the information in a system and establishes the relationships which exist between them. These processes also help to ensure that data is accurately defined. The same name for an item of data may be used in different departments to mean different things (or different names used to mean the same thing). For applications confined to a single department, this creates no problem, but when data items are used to produce information encompassing more than one department, problems can arise. A simple example of this may be monthly purchases which in one department are quoted including VAT (value added tax) and in a second are without VAT.

2.15 Flow

The flow of information in a company has already been discussed. It generally follows the formal organisation structure, i.e. vertically. There is also formal horizontal information flow between departments of a company. The vertical flows provide for the control of the company: performance information going up and policy and plans coming down the structure. The horizontal flows provide the operational information for the company. Little has been said about informal information flows. In a well run, well disciplined organisation there should be little need for informal information if everyone is kept well informed by the formal system. However, there will always be a certain amount of communication outside this. Information which comes from outside the organisation concerning competitors, government policies, etc. is often informal. Not all data can be collected and formalised, and management will sometimes ignore the formal information received because it has heard something 'on the grapevine', or has read a speculative article in a newspaper.

Similarly, within the organisation, a lower level manager may change a decision because they have heard something from a colleague which contradicts information from the formal system. At this level, however, there is more likely to be a defect in the accuracy or timeliness of the formal system than anything else; if it occurs frequently it should merit a formal system review. Also, in this context the analyst has to beware of the 'little black book' syndrome. Managers at a supervisory level, disenchanted with the formal system and its shortcomings, and being unable to get any satisfactory changes made to it, can take the law into their own hands and create their own informal systems, perhaps keeping their own private records in a 'little black book', or on their own personal computer, which is not linked to the corporate systems, and relying on those records rather than those of the formal system.

2.16 Documentation

Finally, it is necessary to look at the various ways in which information can be documented (recorded formally.)

The kind of records which a computer system might hold could include day to day working data such as orders, invoices, details of customers and products. Information may be accessed on-line within the computer system without ever being printed out, or may required in 'hard copy' (printed form). Printed documents include the historical reports, exception reports, and forecasts discussed earlier. Additionally, customer-facing documentation such as invoices are often produced in paper form. Electronic Data Interchange (EDI) takes this one stage further and enables the direct transmission of data to suppliers for ordering and paying for goods using predefined electronic document standards. Catalogues of products and price lists are other forms of routine information which could be computer-stored and produced. A standard form (such as an order form) can be computer produced, used for manual data to be completed, and then used for subsequent computer input when completed. Such a document is often called a 'turnaround document'. The place of such turnaround documents is reflected in web sites, which may provide a partially-complete form to be further completed on-line.

Alongside the corporate computer systems, there are also less formal systems. Office automation, brought about by the wide availability of personal computers, networks and software packages (offering word processing, spreadsheets and database facilities etc.), can handle a great deal of internal communication (both formal and informal). Electronic mail (e-mail) enables the transmission of documents both internally and externally.The minutes of a meeting, for instance, may be typed or word-processed, but held locally on one person's computer, or shared between members of a team via common areas on network hard drives or via the company's private intranet. Minutes are produced to record formally:

- ❑ that a meeting took place;
- ❑ that certain decisions have been made;
- ❑ that those involved are informed of their subsequent responsibilities, i.e. what they must do as a result of the meeting;
- ❑ that those not involved are informed how they might be affected.

Memoranda (memos) and electronic mail (e-mail) are produced by individuals to inform colleagues, subordinates or superiors of some action that has been taken or needs to be taken. Letters, e-mail and fax are also a common means of external communication to or from other organisations. Many communications are routine and can be produced by computer when required: for example a letter to remind a customer of non-payment of a debt, or a general letter to remind all customers of a price increase. For regular communication within the organisation, a notice could be posted on notice-boards or on the corporate intranet or circulated through the internal mail or via e-mail to selective members of staff.

Other types of routine information which could be 'word processable' are instruction manuals and job descriptions. Every employee should have some definition of their job, so that there is no ambiguity in its understanding. Similarly, every procedure should be defined in an instruction manual, to provide training and act as a continuing reference book whenever there is any lack of understanding in the work area.

2.17 Summary

This chapter has covered considerable ground, identifying the various types of business organisations and the principles which govern their effectiveness.

It has considered data and information within a business and looked at the storage and use of corporate data and information. Later chapters will look more closely at data and information, together with techniques for analysing and validating them.

Information is the life-blood of any organisation. It is necessary to ensure that the right information is available at the right level and that it flows to the right people. The analysts, in their development of a system, need to be aware of the information requirements and how the recipient intends to use it. They must ensure that any system developed provides both timely and accurate data and that the appropriate documentation is used.

The analyst should not underestimate the use made of informal information flows, as these can have a significant bearing on the decisions made in a company.

Exercises

2.1 What are the main types of business organisation in the economy of your country? Is this a mixed economy?

2.2 What functional areas (possibly departments) would a typical business have?

2.3 What are the classical principles for successful operation of a business?

2.4 What is the classic view of people's motivation and of organisation structures?

2.5 Compare and contrast the approaches of Maslow, Hertzberg and McGregor to individual motivation.

2.6 How does group behaviour influence individual behaviour?

2.7 What are the major effects of the Human Relations School on organisation structure?

2.8 What is the difference between data and information?

3

Systems and the systems life cycle

Introduction

Within an organisation there are a number of systems which operate in parallel with each other and the interaction between such systems needs to be borne in mind during the development of information systems.

Broadly there are functional, social and political systems within any organisation. A functional system is concerned with what the employees do; a social system with the values people hold; and a political system with their welfare. An analyst is concerned with an information system, but this cannot be developed in isolation from the others. The analyst must be aware of the ramifications of imposing a computer system without due regard to the functional, social and political aspects.

Whilst a computer system should help the user in the functional aspect of the job, it is unlikely to take over *all* functions carried out. The information system must, therefore, interface with manual ones. The information system can also have an effect on the social system, which is concerned with what people think is good or bad about their relationships with others, by changing their roles and responsibilities. Politically, a new computer system may be seen to have an impact on job prospects and status, which could in turn adversely affect acceptance of the new system.

After examining various types of system and how a system model for a typical business enterprise might be constructed, our discussion will focus upon the systems life cycle for the development of computer systems.

In the discussion of the systems life cycle we shall identify:

- ❏ the project stages through which all systems progress; they will be defined and explained, with particular reference to the techniques available. The terminology used is explained;
- ❏ the roles of all participants in the different stages. The importance of the abilities, skills and personal characteristics of the analyst is shown;
- ❏ the relationship between an operational business which has to keep going, and a system project which is separate, and has different priorities and pressures, is identified. The problems which can arise are outlined, and put into context.

3.1 General systems concepts

The Oxford Dictionary describes the word *system* as

'a set of connected things or parts that form a whole or work together.' (Figures 3.1–3.3)

The general model of a system would show inputs, processes and outputs, as illustrated in figure 3.4.

The objectives of a system demand that some output is produced as a result of processing suitable inputs.

A writer inputs words and pictures to produce a Book, as in Figure 3.5.

An accounts clerk produces monthly expense details by processing data about Purchase Orders and Invoices, see Figure 3.6

An individual system has a *boundary* outside of which is the rest of the business environment.

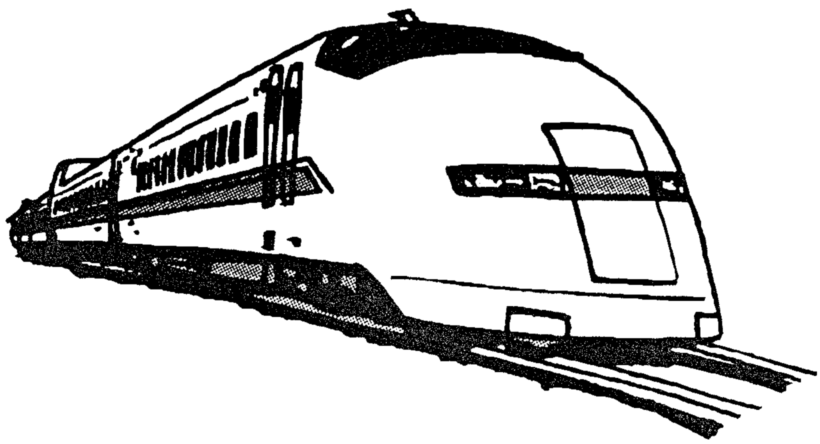


Figure 3.1 A railway system.

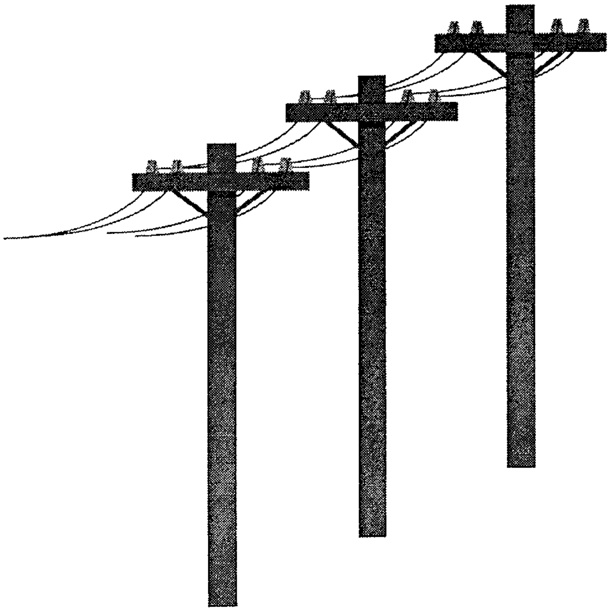


Figure 3.2 A telephone system.

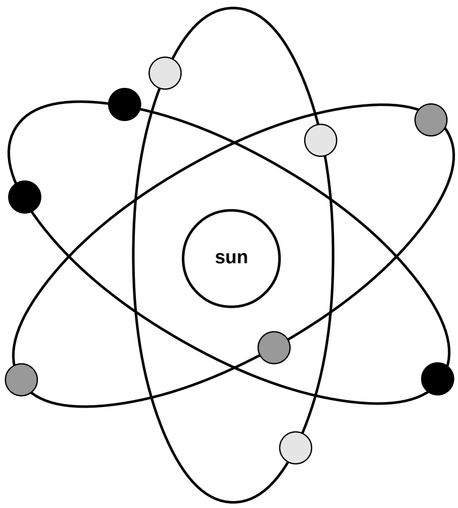


Figure 3.3 A solar system.

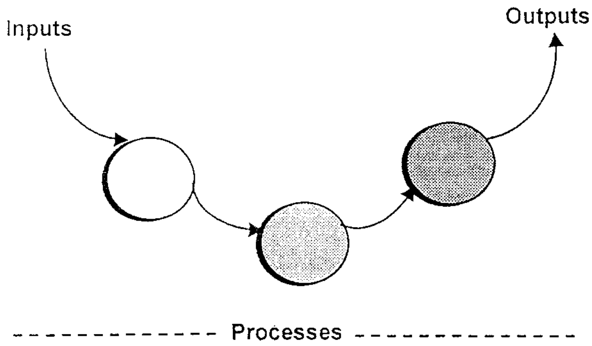


Figure 3.4 Model of a system.

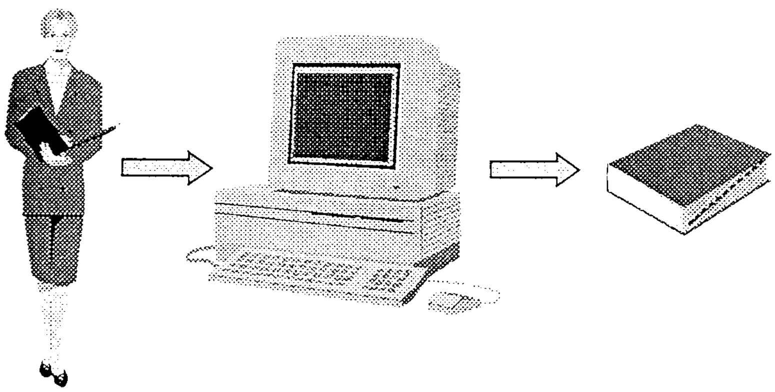


Figure 3.5 A book writing system.

A purchasing system orders goods from suppliers. It keeps a list of suitable suppliers, their products, lead times, minimum quantities, etc. The *inputs* to the purchasing system are requests for parts, the *processing* is to select the most suitable supplier, negotiate rates and arrange delivery times and dates, the *outputs* are purchase orders and notification to the warehouse that the parts have been ordered.

The purchasing system could look like Figure 3.7. Examining the order processing system (Figure 3.8) the purchasing system becomes *part* of the larger system.

Expense code	Description	YTD	Aug.	Sept.	Oct.	Nov.	Dec.
301	Office Supplies	$3,045.00	$500.00	$732.00	$433.00	$732.00	$433.00
302	Office Postage	$1,460.00	$ 45.00	$255.00	$325.00	$255.00	$325.00
303	Office Equipment & Furniture	$ 963.00	$765.00	$ 78.00	$ 21.00	$ 78.00	$ 21.00
304	Miscellaneous Supplies	$ 809.00	$568.00	$ 49.00	$ 36.00	$ 49.00	$ 36.00

Figure 3.6 Expense details.

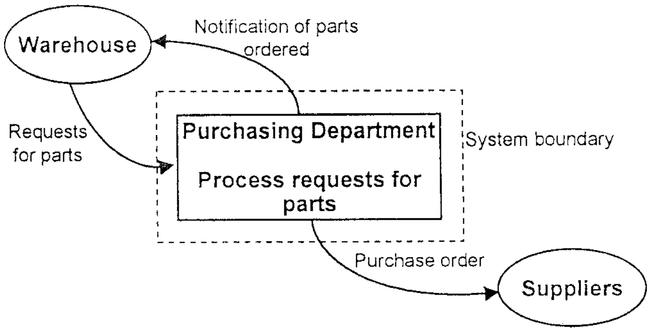


Figure 3.7 A purchasing system.

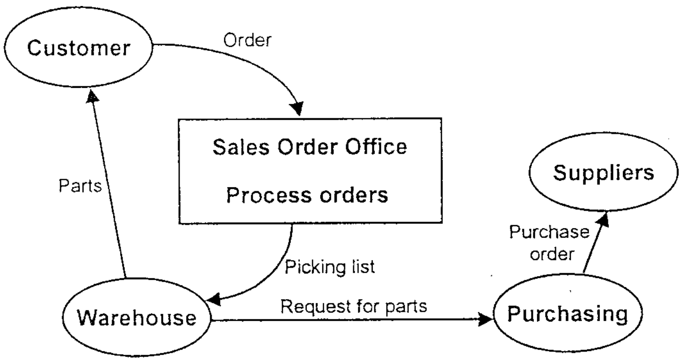


Figure 3.8 An order processing system.

3.2 Categories of system

Each system is composed of sub-systems which are themselves made up of other sub-systems.

Interconnections between sub-systems are often called interfaces; these are made up of inputs to and outputs from sub-systems and clearly each interface is a potential communication path, or data flow.

The system concept requires the observer to examine the system as a whole, but often the total system is too large or too complex for one person to handle at one time. Therefore, it is necessary to divide or factor the total system into sub-systems and then to integrate the sub-systems.

The parts system (Figure 3.9) can be viewed as a whole, within which there are a number of sub-systems (Figure 3.10):

❑ Order Processing;
❑ Accounts;
❑ Stock Records;
❑ Purchasing;
❑ Delivery;
❑ Sales;
❑ Goods Received;
❑ Warehouse.

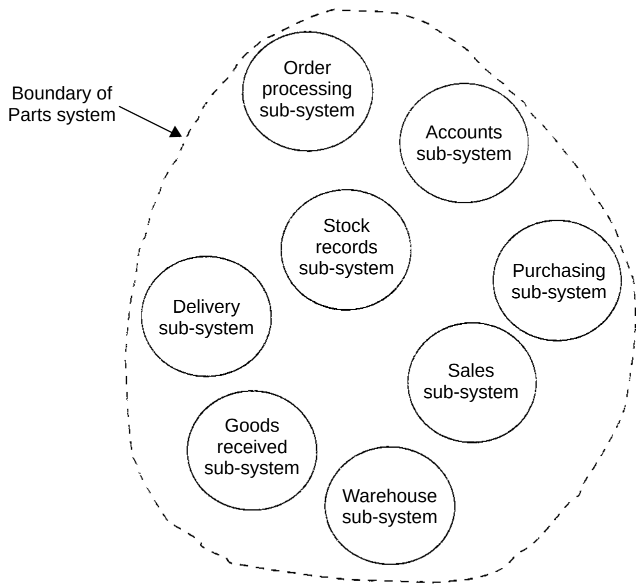


Figure 3.9 Parts system.

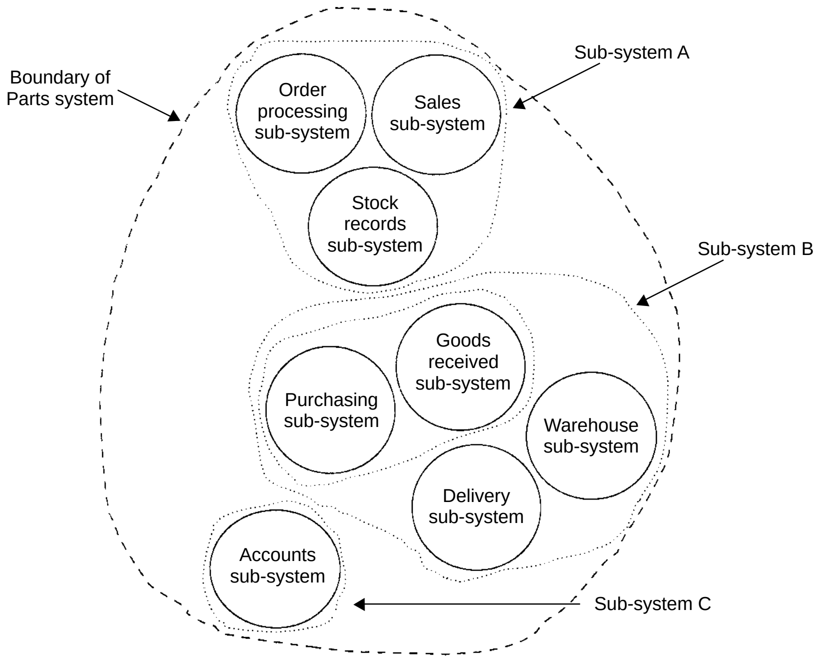


Figure 3.10 Integrating the sub-system.

3.3 Control in systems

Open systems which interact with their environment need some form of control to adjust the level of inputs or the nature of the processes in order to achieve required levels of output in changing circumstances. The control mechanism is known as a feedback mechanism. The control in the system is illustrated in Figure 3.11. The measurement function records deviations from set limits to allow intervention which returns the system to stability. The control may be exercised automatically or by human intervention.

Stability in a system is achieved by having a number of interacting feedback loops, such that if one loop goes out of control another can be activated to restore control.

A control loop in a system can be either open or closed. An open loop is one which interacts with the environment and brings about adjustments in the light of changes in the environment (as in an adaptive system). A closed loop is one which has no interaction with the environment. In a central heating system, the thermostat which controls the temperature of the water can be considered to be a closed loop control because it maintains the water heat

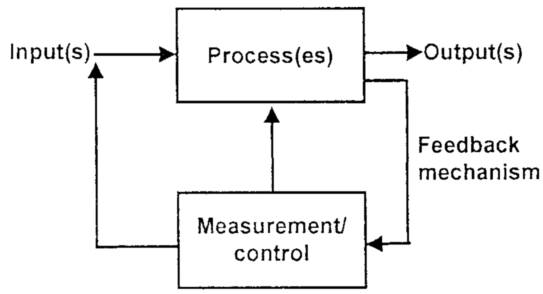


Figure 3.11 Control in systems.

within a few degrees regardless of the room temperature. However, the room thermostat can be considered to be an open loop control because it attempts to control the heating of a room. Of course, this statement depends on whether one views the temperature of the room as part of the system or part of its environment.

3.4 Elements of the system model

The systems approach to problem-solving is basically a way of thinking in which the organisation is viewed as a complex system of interdependent elements, which interact with each other and with the environment. It involves the tackling of sub-problems (or sub-systems) which have been identified as a result of factoring higher level problems (or systems). Thus the emphasis is on the interaction of elements and the need for integration; by looking at the whole system, problems are put into perspective. Normally, the systems approach is carried out in a number of phases, usually identified as systems analysis, system design and system implementation.

The systems analyst/designer deals with systems that are man-made, and therefore capable of redesign. They analyse systems in terms of their objectives, and the inputs, processes and outputs required to achieve these objectives. They are concerned with identifying the boundaries of systems and sub-systems, the interfaces between sub-systems and the factoring into and integration of sub-systems. The systems which they observe are human, open, probabilistic and complex; to adapt systems they must understand how control can be achieved.

Within this overall framework, certain problem areas confront the systems analyst:

- ❑ identifying the system and its boundaries;
- ❑ optimising sub-system interfaces;
- ❑ maintaining the system's dynamic interaction with its environment;

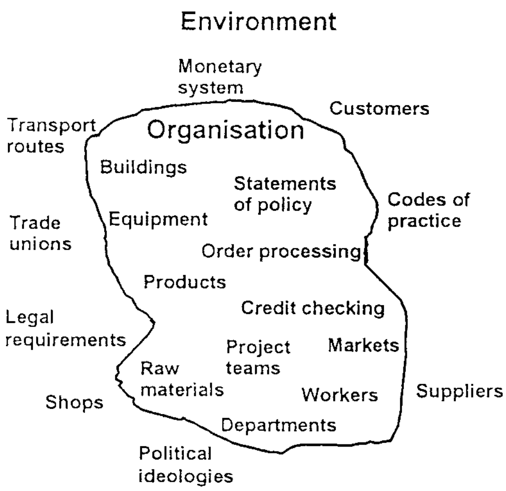


Figure 3.12 The business system.

- ❑ building flexibility of control into systems;
- ❑ resolving conflicting objectives.

An organisation can be studied in systems terms. By definition, an organisation is a collection of people who have a common objective relative to the organisation's existence, though other objectives may be in conflict. Normally, an organisation involves division of work among people, and it is this division that requires 'organising'. The division of work into functional areas provides one identification of sub-systems; one of the sub-systems will be concerned with overall management control.

Now to examine a business organisation as a system (Figure 3.12). The elements of the system are:

- ❑ physical (buildings, raw materials, finished products, equipment);
- ❑ procedural (order processing routines, credit checking procedures);
- ❑ conceptual (statement of policy, market for products);
- ❑ social (workers, project teams, departments).

The system is open to changes in its environment. The elements of the environment include:

- ❑ physical (shops, transport routes);
- ❑ procedural (codes of practice, legal requirements);
- ❑ conceptual (monetary system, political ideologies);
- ❑ social (trade unions, suppliers, customers).

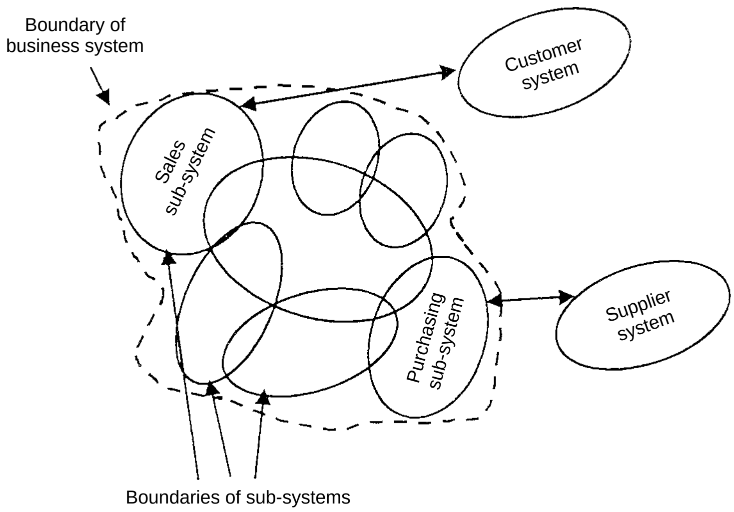


Figure 3.13 Sub-systems of the business.

The business can be factored into a number of sub-systems (see Figure 3.13), normally the functional areas, including:

- ❑ those dealing closely with the environment (e.g. sales and purchasing);
- ❑ those dealing with transformation functions (e.g. production);
- ❑ those acting in a supporting role (e.g. personnel, accounting, information processing);
- ❑ those carrying out overall control (management).

These sub-systems have interfaces with each other and with the environment. Some of the sub-systems have interfaces with systems which form part of their environment. For example, the sales sub-system interfaces with the customer's sub-system *via* customer orders, and the purchasing sub-system with the supplier's sub-system *via* purchase orders.

At the same time as the sales sub-system is interfacing with the environment (customer's sub-system), it is also interfacing with the other sub-systems within the organisation. For example, the accounting sub-system will supply information about prices, discounts, customer payments and customer turnover; the personnel sub-system will recruit employees for the sales sub-system; the production sub-system will receive details on orders and supply information about delivery dates; the management sub-system will seek performance information and determine marketing policies.

It will be clear from this example that the major interfaces between the sub-systems take the form of information flow. If a similar approach is taken

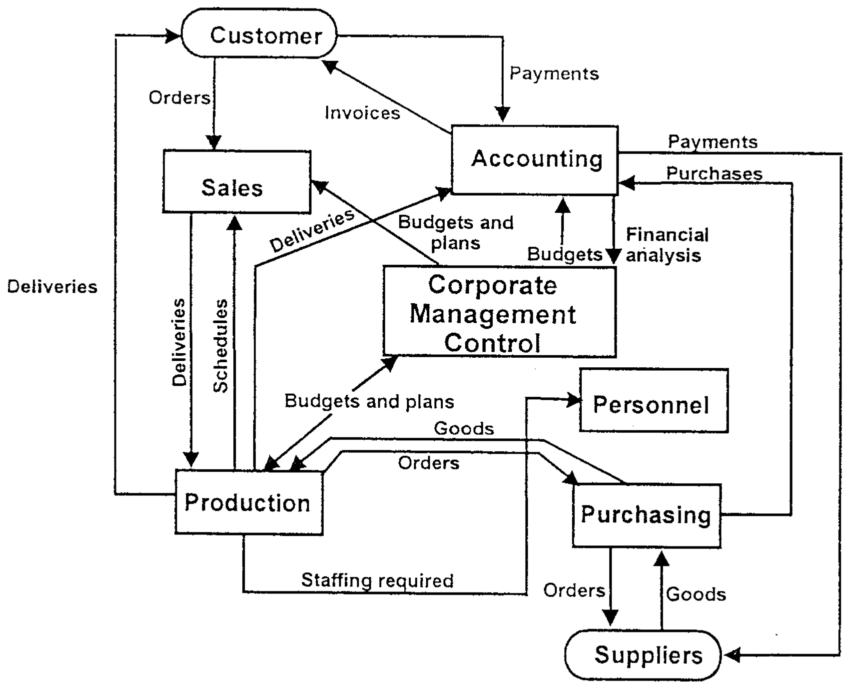


Figure 3.14 Simple model of a business.

to the interfaces between all the sub-systems, a simple model of the sub-systems of the enterprise and their interfaces can be produced. Of course, this does not represent the total information flow, even allowing for its simplification, because each of the sub-systems not only has many more links with other sub-systems and with the environment than shown on the model (see Figure 3.14), but it also has its own internal sub-systems which use information. This simple model illustrates the difficulties of defining boundaries. In some organisations purchasing would be a sub-system of production, rather than a sub-system of the total system. Similarly, distribution is shown on this model as a sub-system of production, where in other organisations it might be a sub-system in its own right, or a sub-system of the sales sub-system.

Each sub-system shown can in turn be taken and broken down into its own sub-systems. These sub-systems have interfaces with sub-systems of other sub-systems of the organisation. For example, if the sales order processing procedures in an organisation are examined, it is usually found that at least four sub-systems are involved: the sales order, sales ledger, invoicing and despatch sub-systems. The sales order sub-system will perhaps be a sub-

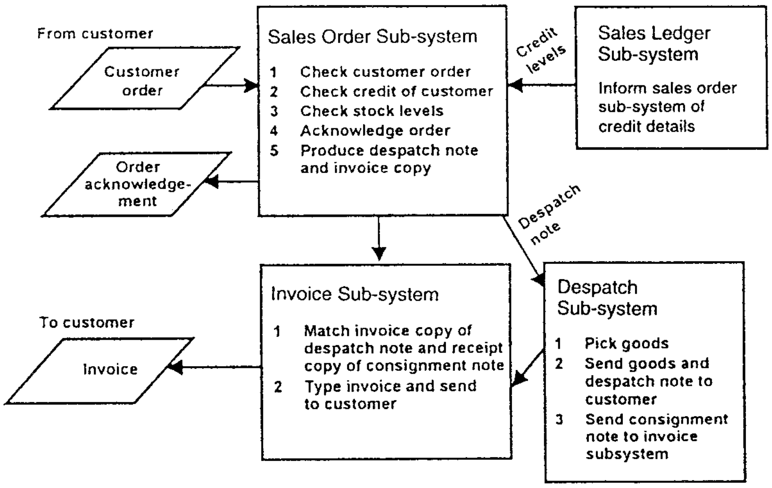


Figure 3.15 Flow of information at sub-system interfaces.

system of the marketing sub-system, the sales ledger and invoicing may be sub-systems of the accounting sub-system, and despatch may be a sub-system of the production sub-system.

Figure 3.15 indicates the sub-systems and the flow of information at the interfaces. Each sub-system could, of course, be seen as a sub-system of the sales order processing system or alternatively of the marketing, production and accounting systems. This highlights the importance of the objectives in determining the system. If the observer's interest is in sales order processing, then the various boxes are sub-systems of a sales order processing sub-system. If the observer's interest is in production, then despatch is a sub-system of production which receives input from the sales order sub-system of the marketing system.

In this example the customer's order is being met from stock, and the sales order office needs a record of whether adequate stock is available to meet the order. Clearly, if the organisation is unable to meet the order because adequate stock is not available, this might result in a lost sale. Thus, the organisation needs a system to control stock levels in an attempt to satisfy the majority of customer orders received. This would be an example of a control system. Normally a stock control system will have two sub-systems: a stock recording sub-system and a stock control sub-system. Stock records are necessary to provide input data to the stock control operation.

Figure 3.16 illustrates the role of the sub-systems. The stock recording sub-system calculates current stock levels by adding receipts to an opening balance

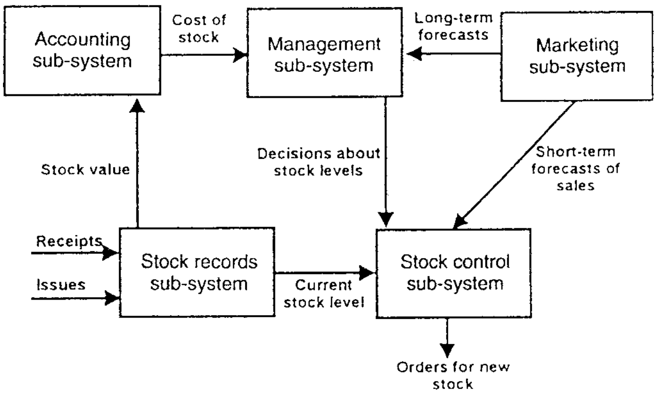


Figure 3.16 Stock control system.

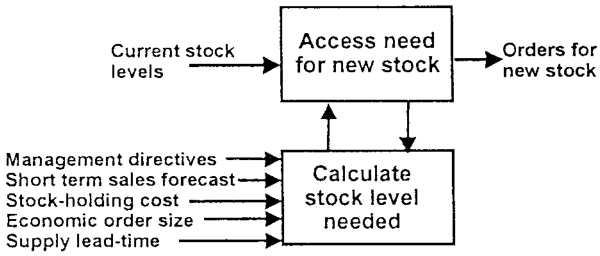


Figure 3.17 Factors in determining stock levels.

and deducting issues; the current stock levels are then input to the stock control system. Also input are management decisions about desirable levels of stock (influenced by cost of stock holding and forecasts of sales) and short-term sales forecasts. These last two inputs enable a calculation of desirable stock level. The current stock level is compared to the desirable level and if an order is required for more stock, then the stock control system initiates the order. This simple system of stock control is, of course, more complex in real life.

Figure 3.17 illustrates some of the more complex factors that need to be taken into account. The desirable stock levels are influenced by management directives, short-term sales forecasts, stock holding cost, economic order size, supply lead-time, etc. The feedback mechanism in this case can be simple or complex. In simple terms it can work by comparing current stock level to desirable level and placing an order of appropriate size if it is below the desirable level. In more complex terms, it can be compared to a number of desirable levels (safety level, danger level, etc.) and a different decision taken depending on the relationship of the actual stock to these levels; and of course the negative feedback may be supplied by human intervention and 'seat of the pants' judgement about what the stock level should be.

Control systems of this kind operate in many parts of the organisation and are characterised by four elements:

- ❏ setting of targets;
- ❏ measurement of performance;
- ❏ comparison against target;
- ❏ adjustment of performance and / or target.

The process of control is continuous and it is clearly important that the control systems are effective in correcting deviations from targets; otherwise money is wasted and delays are introduced. Examples of other common control systems, apart from overall management control, are production control, budgetary control, credit control and manpower control. The organisation's accounts are subject to control by external auditors.

3.5 The database principle

The sub-systems of the organisation so far considered have been concerned with the specific functions of that organisation such as marketing, production or accounting. The business system as a whole can be seen as one which receives resources (capital, plant, people, materials) and produces goods or services.

Each of the sub-systems makes use of information flowing into and within the business in order to be effective. The interfaces between the sub-systems are the lines of information flow. In fact, for the organisation to operate as an effective unit, it needs an effective information system superimposed over and implanted in all the sub-systems. The information system is concerned with procedures for the storage, control and flow of information which is used by, and passes between, the functional sub-systems to ensure that common and accepted approaches are taken to the tasks which are presented to the organisation. The information system provides information for decisions and control and acts as a linking mechanism between the functional sub-systems.

Because of the enormous complexities associated with developing a total information system for the organisation, most businesses have developed a number of information systems rather than one integrated system. Examples of such systems would usually be functionally oriented (production information systems or marketing information systems), or even problem oriented (sales order processing systems, stock control systems). However, it

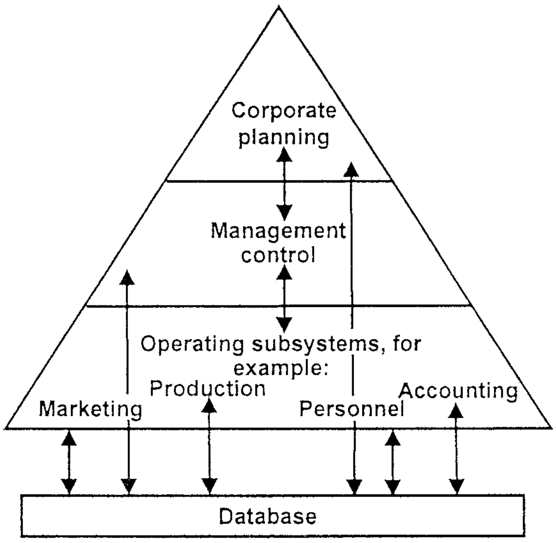


Figure 3.18 The database principle.

will be clear from what has been said earlier that the development of any information system will affect other functional areas as well as the one directly involved, and therefore emphasis has been placed in recent years on achieving an integration of data held on files in order to co-ordinate all information systems. Figure 3.18 illustrates the principle involved.

The database is seen as the store of data needed by the sub-systems of the organisation. The database may be centralised (i.e. held in one common pool) or decentralised (i.e. distributed to separate pools associated with individual sub-systems, but still linked), but its design will take place in a co-ordinated way. The database serves the transaction-processing activities of the operating systems (e.g. payslip production, order processing, sales ledger) as well as the management control activities (e.g. reports on stock levels, sales, cash flow, scrap) and the corporate planning activities of top management. The data is thus available in a consistent and accurate form to all levels of the organisation and this facilitates a totally integrated information system.

3.6 The systems life cycle

When we defined a system as a set of interconnecting parts, together with the interconnections we did not mention of size, nor type of system. Each part may itself be a system. The system described may only be part of a larger system.

3.6.1 The objective of systems work

It is necessary to provide a clear objective for the work of systems people. (They may be analysts, or designers or may have other titles.) This objective could be phrased broadly as follows:

> To produce a full specification of a system which will do what the user requires at a price which they can justify.

Note that no computer is mentioned in the definition. Normally, the user cannot get what he wants unless a computer is incorporated, but the objective does not stipulate that and the analysis could simply lead to an improved way of working. Moreover, the systems objective stops at the 'full specification' stage. Naturally, programs will be required, but their creation is not a direct part of the systems analyst's work.

Use of the words 'requires' and 'can justify' widens the definition considerably. For instance, the user might be quite clear about the essential requirements of their new system, up to the point at which they hear what the cost will be. They may then decide that some requirements are not quite as essential as originally envisaged. Balance between user requirements and cost is necessary during many of the stages of design and development of a system. Since the analyst cannot make these decisions (they can only be made by the user) the implication is that several attempts will be required to establish the system which is required and justified. This iteration will occur many times.

3.6.2 Why a life cycle?

The environment in which any enterprise functions does not remain static for very long, and therefore the enterprise itself needs to keep changing. Because of this the systems, whether manual or computer based, also have to change. Some change is very gradual; a slow continuous progression from one state to another. More often, however, change occurs quite quickly. Changes in the environment have occurred at a greater speed recently than they did in earlier centuries, and therefore no system should be expected to remain unaltered for even a few months.

Some changes can be anticipated and provision made accordingly; for example, expected increases in volume or in types of transaction handled. Others are less easy to predict, for example the increasing importance of the availability of networking systems, even between quite small local processors.

Change is one reason why a system is considered to have a life cycle. Any system will change during the time it is used and will eventually need to be replaced or significantly altered. A system is first conceived, then developed and implemented, then operational until its demise. These stages form a basic *system life cycle* which is defined in greater detail below.

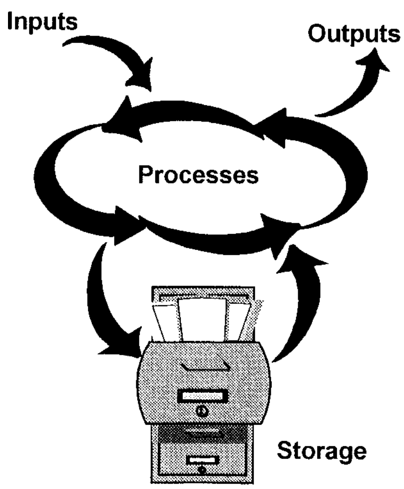


Figure 3.19 Basic characteristics of an information system.

Any systems life cycle model seems to represent the process as a simple, sequential step-by-step development. But this is an over-simplification since the iterative nature of systems may make it necessary to repeat any stage until it is acceptable. Also, since systems work takes time (a reasonable sized system needing perhaps six months or a year to complete), it is necessary to take into account the fact that the environment may alter during development. If the environment does alter, then the terms of reference of the system may also need to be modified (with budgetary consequences for the project, inevitably).

Developing any system requires the analyst to work through a number of tasks. These can best be illustrated by a model showing the flow of the development process. The basic characteristics of any information system (Figure 3.19) are:

- ❑ inputs;
- ❑ outputs;
- ❑ processes;
- ❑ storage (files).

The systems analyst has to identify and define the above aspects in detail. The tasks to be carried out can be shown by different life cycle models. One such model, the waterfall model is shown in Figure 3.20 – the reader is reminded that iteration back (up the waterfall) is inevitable as new information is uncovered and has to be incorporated into the development.

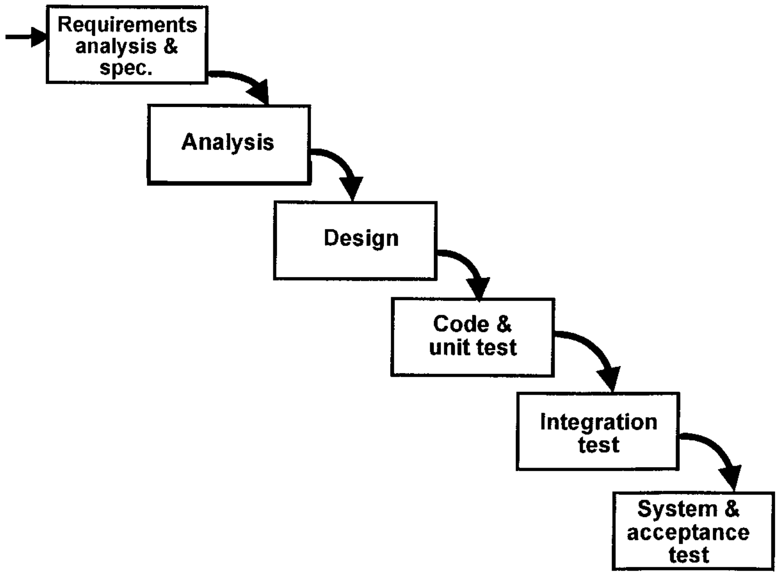


Figure 3.20 The waterfall model.

In spite of the over-simplification, it is still essential for the analyst to recognise the different stages, as different types of work are needed at each stage.

From the initial request and **Feasibility Study** (not shown in the figure but discussed in section 3.7.2) an in-depth **Analysis** is carried out to establish the environment and the exact business requirements, the product from this being the **Requirements Specification**. The analyst carries out fact finding to produce a description of the current system and using this, together with the requirements, builds the logical required system model. Systems **Design** is the next stage, which contains, amongst other products, the program specifications. Once the programs have been written and unit tested (**Code and Unit Test**), **Integration Tests** of all programs running as a complete system are carried out. **System Testing** is undertaken to ensure that everything works in the environment, that is, with other systems, networks, terminals, etc., and that the system is doing what it is supposed to do. **Acceptance Testing** is the users' responsibility, as it is their system and, ideally, they should prepare the test criteria.

The notion of a waterfall in this model suggests that one stage is completed before the next one starts, after all water does not flow up a waterfall. This is a gross mis-representation since in reality it is often necessary to go back to

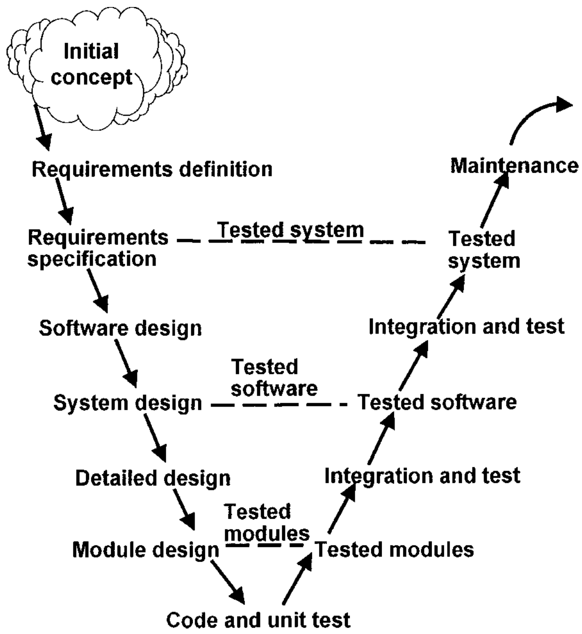


Figure 3.21 The V model.

an earlier stage, so this model is sometimes seen with arrows pointing backwards up the structure to emphasise this.

The *V model* (Figure 3.21) shows the products passed from one stage to the next. These are:

- requirements specification;
- system design;
- module design;
- tested modules;
- tested software;
- tested system.

The advantage of the V model over the waterfall model is that the products are identified at each stage and can be subjected to various levels of testing. Test documentation at each life cycle phase is:

- requirements definition phase:
- requirements specification;
- acceptance test plans;
- system design phase:
- system design;

❑ software test plans;
❑ detailed design phase:
❑ detailed designs;
❑ module test plans.

The final maintenance stage suggests the end of the life of the system. What normally happens is that changes will be required that are more than just maintenance; they may be large enough to require a feasibility study, analysis and design, in which case one returns to the beginning of the life cycle! Systems development is rarely linear, it is usually iterative, as shown in Figure 3.22. This model is sometimes known as the *Spiral* model.

3.7 Overview of the stages in the systems life cycle

The list of stages are shown in Figure 3.22 and each is discussed further below.

Before any new systems work is started, a decision will be made on the **Initial Strategy**. The strategy adopted by an Information Technology (IT) department will be part of, and derived from, the overall company strategy plans. For each project under consideration a series of questions need to be answered. These questions include:

❑ Who will be involved?
❑ What are the boundaries of the area to be studied?
❑ How long should the initial study take?
❑ When will the decision be taken on whether or not to proceed further?

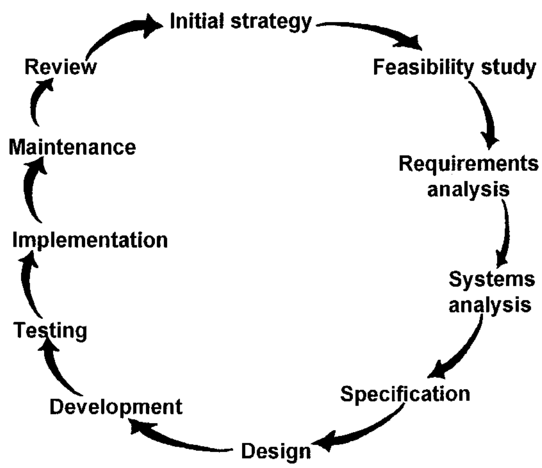


Figure 3.22 Stages in the systems life cycle.

The **Feasibility Study**, lasting a few man-days, will be carried out by an experienced analyst who produces a feasibility report. This describes the scope of the new system and contain estimates of the time, costs and benefits which would result from developing the system. The report will form the basis of the first decision. Detailed work will then commence if the project is to continue.

The next two stages, user **Requirements Analysis** and existing **Systems Analysis**, may be carried out in parallel, but they are quite distinct activities. The user requirements analysis stage will produce detailed information concerning what the users would like the proposed system to do. The result of the existing systems analysis stage is an accurate record of the current system.

The **System Specification** stage involves the production of a complete written (or drawn) statement of what the new system is required to do. There must be no implication of *how* it is to be done, only a statement of what is to be done expressed in a way which is comprehensible to both user and designer.

From an agreed specification, **System Design** can start. This involves a consideration of how the work will be done. As a result, one or more possible systems will be identified and another decision point is reached. The decision to be made is which, if any, of the possible systems will be developed? If development goes ahead, the next stage begins.

System Development includes either in-house programming or the selection and installation of suitable packaged software, and the production of user procedures.

Ongoing and throughout development, and also as a separate stage after development is complete, thorough **Testing** is necessary. This is a good illustration of the earlier comment that stages do overlap each other. Testing is not entirely contained within a separate stage, but follows every step of development with test planning being carried out as early as possible.

After adequate testing, **Implementation** of the system takes place.

A 'live' system (an operational, running system) will probably need **Maintenance**. Errors may be detected which did not show up during testing, and must be corrected. Worse still, errors of concept may appear; that is, the analyst did not quite understand what the user required. If there has been some fundamental misunderstanding between the user and systems people, further systems work will be required.

Any system which has been in use for sufficient time to make it part of normal procedure needs to be **Reviewed**. Costs, achievement of objectives set out in the strategic plans, and even operational ease of use are all part of the review. A report of the review, sometimes called the system audit report, should be produced containing any recommendation for improving the system. Senior management consideration of the requests or recommendations for change will give rise to a new strategic plan, and the cycle will start again.

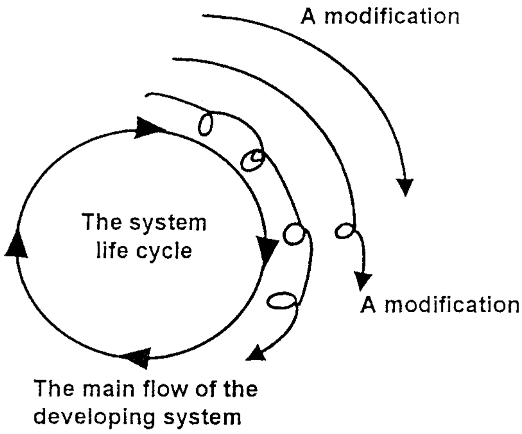


Figure 3.23 The system life cycle – a realistic development scenario.

All systems follow this cyclic pattern. Some may go around the loop many times during their useful life. Alterations, modifications or any other responses to changed requirements also follow the same cycle. Normally, modifications go through the various stages much more quickly than a new system. It is therefore possible to envisage the progress of a system through the various stages as a more active and complex pattern than suggested in Figure 3.22.

Iteration at many stages can cause progress to be delayed for revisions and modifications which are hurried through the earlier project stages to catch up with the main system. Figure 3.23 shows a slightly more realistic pattern of progress, with occasional regression to repeat earlier stages and with modifications being put through.

We shall now consider each stage in more detail, with particular reference to techniques.

3.7.1 Strategy

A major new study will usually arise from the company's strategic plan. Smaller studies will occur as a result of previous reviews. In either case, the first result will be a feasibility study for which terms of reference are laid down by management.

The **Terms of Reference** (ToR) must include:

- ❑ a statement of the areas to be included in the study;
- ❑ objective of the study;
- ❑ the man-days allocated;
- ❑ the date by which the report is to be produced;

❑ any comments relevant to the study;
❑ date and author.

These terms of reference should be approved by the manager whose seniority is sufficient to include the whole of the area under study. They should then be circulated to all management personnel who have responsibility for an area included in the study. This serves to introduce the analyst and also to indicate their reason for being there.

3.7.2 Feasibility study

A feasibility study is an assessment of a proposed information system to determine whether the system can effectively meet the specified business requirements of the organisation, and whether a business case exists for developing such a system. Where a project is low cost and low risk from the organisation's point of view, a feasibility study may not be considered necessary. Larger projects necessitate a feasibility study to determine the practicality of the project and determine the cost justification for the project. A feasibility study requires input from management in the form of a project initiation document which includes the terms of reference and should relate to the company's business and information technology strategy.

The feasibility study involves a short period of observation and interviews during which the analyst (usually an experienced practitioner) conducts a rapid investigation and makes a series of value judgements on the probable scale of system required, and what it would do.

Obviously, the feasibility report is not, and cannot be expected to be, the last word. It is intended to indicate two things:

❑ That a computer based system can, (or in some cases cannot) achieve the improvements which are required. These may be the provision of more information, speeding up the processing of data, reducing system cost, or any combination of these. The precise improvements and benefits cannot be specified in detail, but generalisations are sufficient at this stage;
❑ Second, the report gives cost and timescale, both in development and running of the suggested system and compares this to the benefits to be gained from the development of the new system. This is formulated as a cost justification using Cost Benefit Analysis. On the basis of the feasibility report, further work on the system can be authorised.

3.7.3 Cost justification

Cost justification is a process of identifying financial costs and benefits associated with a development project. Cost benefit analysis takes place at various points in the development life cycle such as the feasibility study and

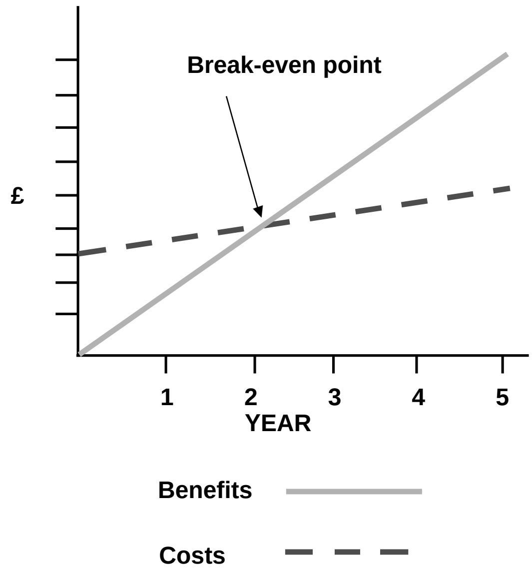


Figure 3.24 Graphical representation of cost/benefit analysis.

when making a choice from a number of business and technical options. Non-quantifiable or intangible costs and benefits may also be included.

The purpose of cost benefit analysis is to measure the costs associated with the development of the new system and to compare this to the benefits associated with the development. Tangible costs/benefits refer to items which can be measured in currency terms whereas it is difficult to put a monetary value upon intangible costs/benefits.

On the costs side of the development, tangible costs include developer and user staff time, hardware and software purchase, site preparation, training and data take-on, implementation and operating costs etc. Intangible costs are such things as disruption to existing systems, possible temporary loss of goodwill and adverse effect on employee morale.

On the benefits side tangible benefits include reduction of costs and level of errors, improved processing speed, better planning and control and overall improved efficiency i.e. better quality and quantity of production per member of staff. Intangible benefits encompasses competitive advantage, better management information allowing quicker decisions and responsive planning, better staff morale etc.

The cost/benefit analysis is best represented as a graph plotting a line showing costs and a line showing benefits against time. (Figure 3.24).

The output from feasibility study is a *feasibility report* and is the basis upon which a project may (or may not) be initiated. The report will include outline business and technical systems options which will be developed further during analysis and design.

3.7.4 System options and constraints

System options influence the decisions upon whether and how a project will move forward. The system options can be divided into:

❑ business system options;
❑ technical system options.

Business system options

Business system options are the mechanism for the analyst and the project team to inform the project sponsors of the alternative ways in which their system can be developed to meet their requirements. This will allow the sponsors to make informed decisions on the way ahead.

Development of business system options gives analysts and users the opportunity to explore where the system boundary lies. It will provide the analyst with a starting point for the specification of the required system and the users with an initial overview of the system they will be receiving.

It should include:

❑ system boundary including a description of all the proposed functionality;
❑ levels of functionality, including at what level the entire application and its components are intended to function;
❑ cost/benefit snalysis;
❑ impact analysis on existing information systems, the infrastructure and the business area;
❑ technical considerations of the different options;
❑ base constraints.

Technical system options

Technical system options are developed as an elaboration of the selected business system option. They will address:

❑ a specification of the technical environment, e.g., provision and distribution of hardware devices, software environment, operating regime;
❑ confirmation of the functions to be covered, and the manner in which they are to be carried out;
❑ the impact of changes to the organisation and methods of working;
❑ impact on the development organisation and infrastructure for the remainder of the project.

Technical system options is the mechanism by which the project manager provides the user management with specific technical information on the way forward, its costs, implications and timescales. User management will then make an informed decision, selecting the most appropriate way ahead related to their organisation and the project's objectives.

3.7.5 Requirements analysis

The information gathering activities associated with requirements analysis and existing systems analysis are normally carried out at the same time. Requirements analysis, however, will include everything gleaned from interviews and observations which does not refer to the current systems. All ideas concerning how the work could be done more adequately, or about what ought to be done better or differently, are gathered and recorded during the analysis stage. All this information is used with the details obtained during analysis of the current system, to define the new user requirements.

3.7.6 Existing systems analysis

The analyst has three tasks during this stage:

- ❏ To understand the existing systems, both in detail (*what* is done) and in principle (*why*);
- ❏ To record this information in a way which they and the users can understand and agree;
- ❏ To refine the recorded statement in a logical (rather than physical) specification, and add the requirements for a new system to it.

At this stage they are analysts, not designers. Indeed, the analyst and designer may be different people. However, the reason for stressing the distinction is to make it quite clear that no design decisions are made at this stage.

The tools which the analyst uses to record what they find are Data Flow Diagrams (DFDs), miniature specifications, various data analysis methods and a Data Dictionary (DD). These are the subject of later chapters in this book. Using a levelled approach and structuring their records appropriately, the analyst can construct diagrams and data records which the user can understand and, therefore, correct if required. An initial set of diagrams based on who (or what) department or section actually performs various tasks is the physical specification. Refining this, adding more detail and generally moving away from who does a task to what is actually done, the analyst can produce the logical specification. This is a model of the existing system simply showing what the system does, not how it is done, how it might be done, or by whom.

At completion of this stage the analyst has a logical model of the existing system which has been agreed and corrected by the user. They can change and add to this in order to include the user requirements. They will need to refer to the feasibility report to assess whether all additional requirements which were identified at the earlier stages have been added, and if not then they must add them. This new logical model of the required system must be agreed with the user.

3.7.7 System specification

To produce the system specification, the analyst needs to complete the new logical model. Using analysis techniques, they will have sets of levelled DFDs and a set of miniature specifications to support the lowest level DFDs. They will also have a Data Dictionary (DD) containing all information currently available about the data within the system. For each data flow, there is an entry, possibly referring to other entries. If data normalisation has been carried out, then there is reference to data structures in 'third normal form'. There may be entity models and other related details.

This specification, agreed with the user, will be the basis for the designer's work. It contains both the broad outlines and the fine details of what the new system is required to do. Nowhere does it specify how it will be done, which is a design decision. It should also contain all desirable features of the new system, although not all will necessarily be included in the installed system, as some features may not be cost-justifiable.

3.7.8 System design

At this stage the designer can begin to use the system model to indicate how the various tasks can be done. There are often several options. A minimal computer-based system, with most of the work still done manually, may offer the cheapest solution, perhaps incorporating only the most important of the new features. A massive computer-based system may be needed to include all features. The designer is usually able to present the user with a choice, and with the relevant estimated costs and benefits. This stage, like many others, is very much an iterative stage, and several different systems may be considered and discarded before a final decision is made. Once the decision is taken, the extent of the computer and manual systems are fixed, and the next stage can begin.

3.7.9 System development

Just as the analyst uses structured methods in the early stages of the cycle, so the remainder of the IT development team will use their own structured approaches. Development involves the final definition of data structures and the physical storage method or data management software which will retain and maintain the data, together with the provision of all necessary programs. Testing is shown as a separate stage, but in fact, all development work is tested as it proceeds, using walkthroughs, inspections and direct testing via the computer. Programming aids such as test harnesses may be useful tools.

3.7.10 Testing

As well as the commonly understood methods of testing, there are many other ways of ensuring that the system performs as required. For example, it may be difficult for users to appreciate exactly what a particular human/computer-interface will be like in practice, without seeing it. Some form of prototyping may therefore be used at quite an early stage in the design and development of a system. A trial set of routines may be produced to simulate the interface at, for example, a screen. These can be altered at the user's suggestion, and eventually agreed. The agreed interface can then be written into the final system.

It may be that some (or all) computer programs required can be purchased off-the-shelf. Even so, they still need to be tested as part of the whole system.

3.7.11 Implementation

Normally, a system is changed over in stages. This may be a *pilot* scheme, in which the whole system is implemented in a small part of the enterprise. When that has settled down, more parts of the enterprise are introduced so that, over a period, the system is implemented in the whole area affected.

Alternatively, a *phased* scheme may be adopted. Part of the system which can 'stand alone' is put in over the whole area. Then another phase is added, and so on. It is usually necessary for the system to be designed in such a way that the 'phased' approach can be used, if that is a desirable method of implementation. It has the advantage that all users are working to the same methods all the time.

It is always possible to take the 'overnight' approach, so that the complete new system applies to the whole area affected at a certain time. This may be necessary if there is no suitable pilot area, or if the system cannot be phased.

In any of these cases, *parallel running* is advisable. For a period of time, often a month or more, the old system is kept going with the new system working as well. The results of the old system are still used, as though the new system were not there. The results of the two systems are compared, as far as possible, and when the new system is deemed to be producing adequate results, the old system is stopped. During this implementation stage, it is necessary to have special changeover instructions and procedures for both user staff and computer operations staff.

3.7.12 Maintenance

There are two types of maintenance work, the 'emergency repair' and the 'requested enhancement'. Often, enhancements can be added together and early ones delayed until a new release of the system can be made after some months. Emergency repairs may be required as a result of program error, hardware fault, or even user error. It is important that adequate provision is made to deal with errors, and to reinstate any faulty files which may have resulted. Back-ups are covered in more detail in the Business Systems Design book in this series.

3.7.13 Review

Once a system is installed and working and emergency work is being coped with, the system is 'live'. At this point, an evaluation review should be carried out by an individual who has not been a participant in the design and development stages.

The review should cover costs, comparing the actual costs with those anticipated at various design stages. It will assess to what extent the expected benefits have been achieved or are likely to be achieved. There must also be serious consideration of system performance. The users, now they are familiar with the system, may be able to identify aspects which could be improved. Even such aspects as over-detailed output (which could be needed during the familiarisation period) may be an irritant once the system is running. Users, operations staff and data preparations staff (if appropriate) may all have valid comments.

The review should contain recommendations for the future. These may be ideas for minor modifications, or the need for a complete re-think and re-design, as a result of the changing environment. Reviews should be repeated at least annually, preferably more often. This will bring any need for change to the attention of the management in time for plans to be made. The recommendations, of course, lead directly to the strategic planning necessary to initiate further work, which completes one life cycle and potentially initiates the next.

3.8 Rapid Application Development (RAD) and incremental development

Rapid development takes a different approach. The objective is for a subset of the system to be working early on. This is then gradually built on, so that the system is built *incrementally* (Figure 3.25). The principle is to build and implement the parts of the system which will deliver the biggest business benefit first and build other parts of lesser benefit later on in the project.

The requirements phase involves analysing the user's requirements and specifying a system to satisfy those requirements. The design phase provides

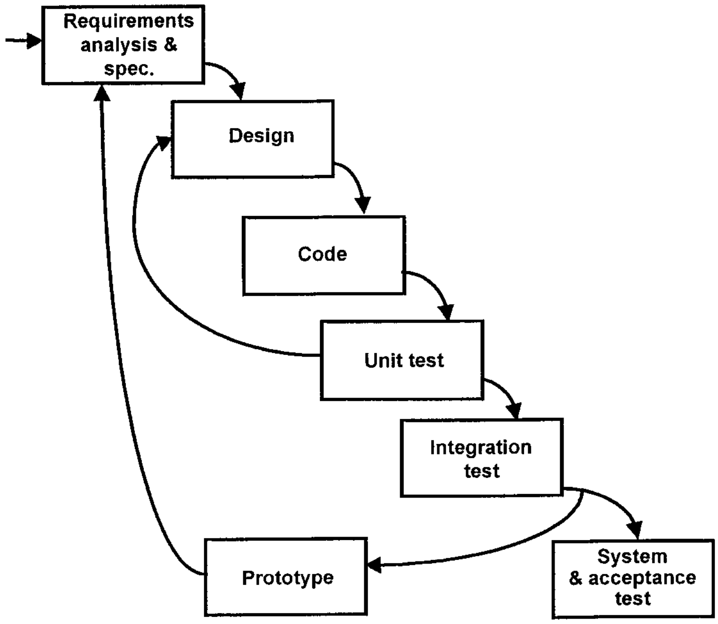


Figure 3.25 RAD incremental system development.

the complete detailed design of each module, which is then coded and tested. As the modules are completed they move into the integration phase, which has a loop back to the requirements phase to allow for changed or new requirements. A completed and tested module, or set of modules, may be suitable for demonstration and acceptance by the user.

Further discussion of a particular RAD approach will be covered in Chapter 4.

3.9 People involved with systems work

There are five categories of people who are involved, in one way or another, with systems work:

- ❑ users;
- ❑ analysts;
- ❑ designers;
- ❑ programmers;
- ❑ managers.

Managers, analysts and designers are the categories which will be discussed in most detail here, but others will be mentioned briefly.

The analyst needs to keep in mind the fact that users, at all levels, have a job to do. The amount of time which they can allocate to describing their work, checking that the analyst has correctly understood it and generally working with the analyst, is likely to be very limited.

Also, users have their own jargon, and often see no reason why they should have to explain it, or indeed do not recognise it as jargon.

The analyst should bear in mind that their own vocabulary should be as jargon-free as possible when talking to users.

Users appreciate their function in the existing system, and often consider it to be obvious. However, it may not be obvious to the analyst, whose care and tact will be necessary to obtain the relevant information.

Analysts are often viewed as the instruments of change, even though their presence is only a symptom of the changes which are occurring. Some users, particularly (though not exclusively) at the lower levels, see the analyst's work as a threat. This again means that tact is needed.

When any new system is finally implemented, it almost always presents users with extra work during changeover and parallel running. Computer people deal with machines for at least some of their time, but they should not forget that users are not machines.

It is very important that an atmosphere of mutual respect should exist between the users and the IT person (analyst,designer etc). The IT person must recognise that the users are the experts in the business. Likewise the users must recognise that the IT staff are experts in the development of computer system. Users need to be allowed by management to put in sufficient time to explain what the current system does and what it will need to do in the future.

3.9.1 Managers

The type of manager the analyst will deal with are:

❑ managers within the project;
❑ user managers.

Managers within the project These can be categorised as follows:

❑ project managers;
❑ configuration managers;
❑ quality assurance managers.

Project manager The project manager can come from the business or IT area. He must understand the business issues, the technical issues and user views. The project manager must produce a system on time, that the users accept and within budget. Their specific responsibilities are project planning and scheduling, targeting and motivating the project teams, setting team objectives, exception handling, managing user involvement with the development teams and handling problems escalated from the project teams.

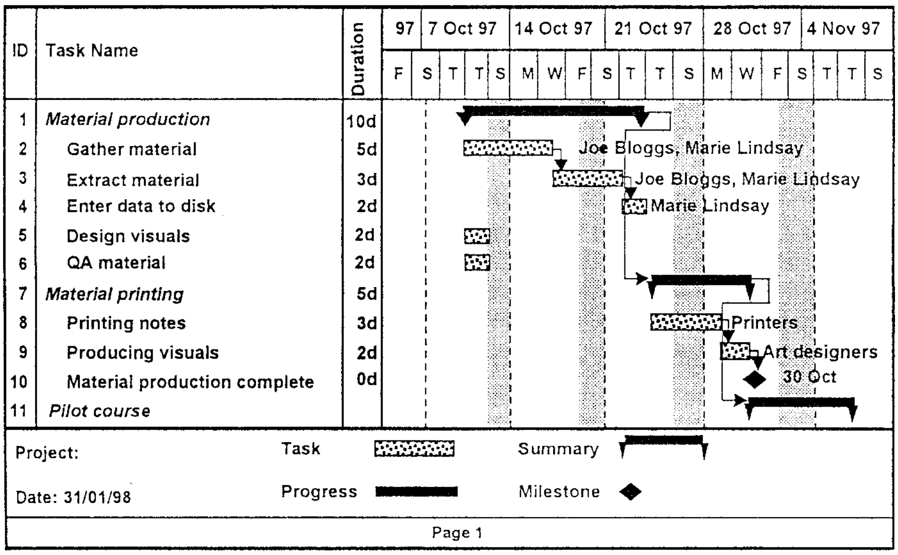


Figure 3.26 Project plan

Figure 3.26 shows a project plan preparing for and running a pilot training course.

Configuration manager Configuration management involves the controlled development and release of software such that its content and status at any stage is known. This applies throughout the development life cycle and continues once the product has been released. It should also include training materials, test scripts and documentation. For example, configuration items for a new version of an existing system could be:

- ❑ the run-time system;
- ❑ the compiler;
- ❑ the compilation macro;
- ❑ the run-time macro;
- ❑ utilities;
- ❑ development tools;
- ❑ new release documentation;
- ❑ training aspects.

This information is placed under configuration management whilst the new release is being developed so that an earlier version can be returned to if there are major problems, as well as when the release is issued.

Quality manager The quality manager is responsible for a *quality management system* which consists of quality control, quality assurance and quality management tools, techniques, methods, processes, practices, standards and guidelines.

Quality assurance ensures that quality standards, guidelines, practices, methods, etc. are all in place and are *used*. The products within an IT system are subject to quality control, which is the *measuring* of the quality of the products to ensure they reach the required standard. This can be achieved by inspections, reviews, testing and tools such as static code analysers, if avalable.

User managers Managers are often users, in the sense that the systems under consideration fall within their jurisdiction. However, as the analyst moves up the levels of hierarchy, they should expect different treatment, and different information. Managerial levels of staff do not normally concern themselves with the minutiae of operating the business, but are directly involved with translating overall strategy into tactics and working decisions. The overall strategy is a high-level decision. Making that strategy work is the responsibility of successive levels of management. Managers direct the activities of those over whom they have authority, and when a decision is required, they make it in the knowledge of the strategic intentions passed down to them. Therefore, it is from management that the analyst is most likely to obtain the correct user requirements. They will be correct because they will, or should, be described in the context of the overall organisation strategy. At lower levels, user requirements may only relate to day to day convenience, or to quite inappropriate ideas of what should be done. However, management may not appreciate that relevant changes could have been introduced into their systems, resulting in a considerable improvement to the original.

In seeking information and opinions from users and management, the analyst should be aware of the possibility that they will occasionally be misled deliberately. Sometimes an individual will see the analyst as someone who, if deliberately supplied with incorrect information, will further the political or industrial aspirations of that individual. This is not easy to guard against, or to detect, but if analysts cross-check their findings with other users' managers and other analysts, they will avoid the worst mistakes.

Programmers are, in one sense, the opposite of users and managers. They are seldom involved in office politics, they understand and speak the same language as the analyst, and they tend to follow very precisely whatever specification they are given. Although this is desirable, it does place a very heavy responsibility on the analyst and designer to ensure that the specification is both unambiguous and correct. Programmers are seldom application experts, and not knowing the fine detail of a particular application will be unable to correct an analyst's or designer's error.

3.9.2 The analyst and designer

Up to this point it has been assumed that the analyst and designer are two different people. Sometimes that is the case, but in other circumstances one person is responsible for both jobs.

The individual will, of course, need training. Certain personal characteristics are also required. They must be up to date with computer technology and have a general understanding of business systems. Specialisation may be appropriate, even necessary from time to time, but the analyst or designer should take every opportunity to keep up with developments and current practice in both business systems and computing, particularly in computer applications, where the two meet.

One skill which the analyst particularly needs to acquire is that of communication. Clear communication between analysts and users, and analysts and management, is essential if systems are to be produced which contribute to the profitability of the organisation.

Finally, the successful analyst or designer should display personal characteristics similar to those outlined below.

They need the intelligence to grasp the fundamental logic behind any methods or procedures. They need to be perceptive, to see what they are looking at, and understand it. Naturally, though perception is important, they should avoid jumping to unjustified conclusions.

They must be able to plan their work and the work of others, to ensure that time and other resources are used most effectively. They should be able to recognise when a plan needs to be altered to cope with some new facts.

To keep to a plan, they will need persistence; they will meet obstacles of many kinds, and difficulties which must be overcome.

The acquisition of skills and knowledge about their specialisation will help to give an analyst or designer the confidence necessary to carry them through the difficulties they will encounter. Negotiation can only be satisfactorily completed with self-confidence and genuine enthusiasm. Similarly, stamina and strength of character are very desirable characteristics in anyone who is likely to be regarded, from time to time, as a threat.

If the analyst or designer can bring a sense of purpose, without being too blinkered, an orderly and self-disciplined approach to the job, and a logical neatness to the way they work, they will clearly work more effectively.

All the communication training in the world will not help someone who is unable to listen. Communication between people is not a monologue, at least it should not be!

Lastly, and in some ways, most importantly, the analyst and designer need to display a professional integrity. If they promise something, it must be done. If they are seen to be underhand, unreliable, or incomprehensible, they will lose credibility very quickly. The analyst must keep in mind that their job is to create systems which make profit for the organisation.

3.10 Business *vs* systems projects

One of the aspects of systems work which should be stressed is that a systems project has certain priorities and a defined progression from initial strategy to implementation and review. As far as the analyst or designer is concerned, each job is new and different, and once one project is 'live', the next one absorbs their interest.

In business, however, continuity is the keynote. For the whole duration of a systems project, the business systems must be kept running with the minimum of disturbance and the maximum efficiency. This is often at a variance with the priorities of the project, but is more important than arbitrary deadlines or priorities within the systems team.

Any work with users which the analyst or designer does, is a disturbance to those users. It may be unwelcome, and often an analyst cannot obtain either the interviews or the information they want, when they want them.

The user will also have a natural tendency to avoid involvement with computer people. This can be a form of self-protection. If the user does not tell the analyst their detailed requirements, and the analyst fails to fulfil them, the user can claim that the computer is useless. It can also arise from pressure of work or from a feeling the user has that they 'cannot understand computers', and will not even try to. In these and other cases, the analyst must do their very best to get the user, particularly the user management, involved with the new system. If the user can identify with it, can see aspects that are their own ideas (or which they think are their own ideas) then the analyst and designer are likely to have a much easier time, and will produce a much more effective system.

3.11 Summary

The concept of an information system as a separate sub-system of the organisation (alongside the marketing, production, purchasing sub-systems, etc.) highlights again the problems which systems theory presents to the systems analyst. An information system, like any other system, will have conflicting objectives, a boundary which is difficult to define, awkward interfaces with other information systems, a requirement for flexible control patterns, a need for built-in dynamism and adaptation, and above all, human interaction which needs to be optimised.

By using some of the concepts of systems theory, the systems analyst has a framework with which to approach the analysis of organisations. In this approach an attempt is made to study the organisation as a whole and its environment, with special emphasis on channels of information flow. The system under observation should not be too wide for the purpose (this would be wasteful), nor should it be too narrow (this could result in an efficient part at the expense of an ineffective whole). The interdependence of sub-systems (which necessitates the systems approach) should be a major concern of the study, but equally important is the need to determine the relevance and effectiveness of existing sub-systems. The assessment of the relevance and effectiveness of a sub-system involves the analyst in starting with the objectives of the overall organisation (these may or may not relate to the current organisational structure). Grouping of decision areas in accordance with information needs is the second stage; the final stage is to design an information system which meets these information needs by providing effective information flow.

Every system introduced will last for a finite time before it is out of date and needs replacing. This is the life (life cycle) of a system. It begins with a strategic decision, goes through various stages of development until it emerges as a live business system. The live system is maintained and reviewed periodically. Eventually, following a review, management will decide that a better way must be found to do the job and thus the cycle will begin again.

It is the live business system which keeps the enterprise operating on a day to day basis. The project to replace it (system project) is to the systems analyst a high priority; to the user it is only of secondary importance. The analyst must therefore be aware of the environment in which they have to do their job and also that of the people with whom they deal.

In later chapters, the emphasis is placed on the system project. The analyst and designer should not lose sight of the fact that the whole of their work is directed towards one single end: a system which increases the profitability of the organisation either by cutting cost or improving efficiency, or both.

Exercises

3.1 What is a system? (Give three specific examples of uses of computer systems in an organisation and describe control loops.)

3.2 What is a database? What are the advantages and disadvantages of having one corporate database containing all of the organisation's information?

3.3 What is the systems life cycle (use diagrams to illustrate)? How does the 'V' model differ from the waterfall model? How is this different from a RAD life cycle?

3.4 What is the role of the systems analyst? What are the necessary contents of the analyst's terms of reference? Who should approve these?

3.5 Who are the people involved in the systems development process?

4

Methods for system development and the role of the systems analyst

Introduction

In the early days of system development, the approach was not well defined and all too often the system, which was delivered, did not do what the user wanted. As a consequence, confidence in the IT department was damaged. In parallel with the development of a structured approach to computer programming, it was recognised that a structured approach to the investigation and development of new systems was needed. This would also require the development of techniques to support the approach.

The concept of a system life cycle, as described in the previous chapter, provided the framework upon which the structured approach could be overlaid. The development of a structured approach in the eighties has given rise to a number of methods, some government sponsored (UK SSADM), others commercially driven (Information Engineering, Prism, Method 1). Since that time, the methods have been refined and Object Orientation has also been developed. Software tools to support the approaches have likewise arisen.

The systems life cycle and defined methods and techniques for the development of new systems have gone some way toward solving the problems of systems development and have taken it into a more disciplined engineering approach.

This chapter will discuss some of these approaches and the role that the systems analyst plays in the development of new computer systems.

4.1 The costs and benefits of introducing a method

Introducing a systems development method costs money in terms of both the direct expenditure on reference manuals, software and training, and the fact that information systems staff will not be able to carry out other duties because of the need for training. An organisation adopting a method needs to plan and control its introduction in the same way as other projects. Success depends on the commitment of management, both to the method itself and to the project development infrastructure (project management, quality management, etc.). There must be commitment to:

- ❑ education in the method for systems development staff at all levels;
- ❑ training in the basic analysis and design skills of the method;
- ❑ training for practitioners and users;
- ❑ providing automated tool support for the method;
- ❑ adapting project management procedures and installation standards;
- ❑ quality management of the use of the techniques in the method and the products.

The first stepping stone in adopting a method is to have a firm and realistic grasp of what it is *not*. It is not a panacea that will abolish all known problems at a stroke, neither is it a means by which improved systems can be built automatically. An outline plan for the introduction of a method is given below.

4.1.1 Obtain corporate approval and commitment

There are costs involved in purchasing the method, purchasing tools to support the method, training, expert assistance and the learning-curve period when other projects may have to be delayed. If innovation is to achieve its full potential success, the support of senior management is essential; without commitment from the highest level results will be poor.

4.1.2 Select a pilot project

Once a decision has been made to use a method, a project should be selected that is suitable for a pilot use of the method.

4.1.3 Arrange staff training

Once a project has been identified, the training of the information system staff who will carry out the analysis and design can be arranged. Training too early is not useful since it can lead to people forgetting some of their newly acquired knowledge before the project starts.

4.1.4 User awareness training

Users should be briefed on the method's products with which they come into frequent contact and told what is required of them. This entails a clear exposition of the amount of time and effort they will now be required to devote to systems development. Some methods require continual user involvement throughout analysis and design, so any problems of availability must be resolved by the line of user management.

4.1.5 Plan for expert assistance

The concept of using an expert or consultant for guidance should be given serious consideration. It may be that the method chosen needs to be tailored to the organisation. Tailoring should only be done by someone with a profound grasp of the totality of the method to ensure that the process does not introduce compromise or corruption.

4.1.6 And finally . . .

The success or failure of the pilot will be increasingly evident once the half-way point has been passed. This is the time to start planning ahead, by training more staff and selecting the next system for development. This is also the time to think about a wide range of strategic matters, which vary from organisation to organisation. An issue that should be addressed is the range of possible problems that will stem from the conversion to the method. These should include systems currently in development and maintenance of all that has gone before. Each information systems department must find their own solutions: the most important thing is to ask such questions and think about the ramifications.

4.1.7 Benefits of a standard approach

The adoption of a standard approach followed by everyone involved with systems analysis within an organisation yields many advantages:

- ❏ *increased productivity:* the required tasks within each stage of analysis are clearly defined so that each analyst does not have to 'reinvent the wheel';
- ❏ *better quality systems:* structured techniques prompt the analyst to ask the right questions and will highlight areas where information is missing. They include review checkpoints during all stages of system development, the purpose of which is to check for and remove any errors that may be present. They also encourage a modular approach which leads to more flexible, maintainable systems;
- ❏ *better communication:* the techniques, which are mainly diagrammatic, can be understood by both analyst and user;

- *better project management:* the definition of tasks within analysis, together with standard techniques for their performance, makes progress during analysis more visible and therefore more controllable;
- *better documentation:* the structured diagrams which aid analysis are also the main documentation of the system. Thus documentation is produced hand-in-hand with investigation and development.

4.2 Stages in analysis – a generic view

The processes of analysis to which the structured methods apply is now discussed generically and shown in Figure 4.1. This corresponds to the systems analysis phase of the system life cycle.

1) The first stage, fact finding and analysis of current system, will subsequently be referred to as investigation – the analyst starts with terms of reference and has access to historical records. The results of interviews form a major part of this stage, allowing the analyst to build up a model of the existing system. Each movement of data is represented on one of a set of DFDs, as are data stores (files). The result of this stage is a physical description of the existing system which will include sufficient data dictionary entries to identify the named data structures. The DFDs produced during this stage are often called *current physical DFDs.*
2) The second stage is the derivation of the logical system, removing the 'who', 'where' or 'how' details to leave a statement of 'what' is done. Normally,

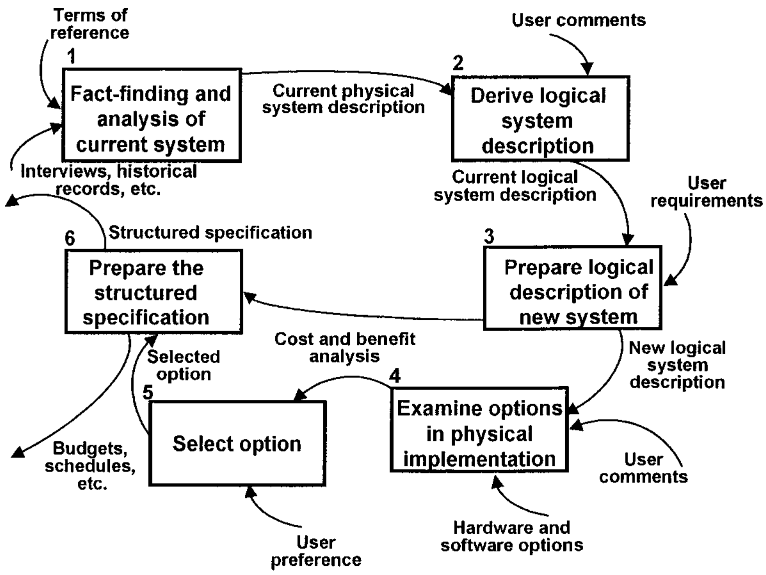


Figure 4.1 Stages in analysis as applied to structured methods.

two, three or at most four levels of DFDs are needed, as will be explained in subsequent chapters. There will almost certainly be several attempts required to achieve the logical description. User involvement is also needed to ensure completeness and correctness. Entity analysis is carried out to identify the main logical groups of data and a data model produced. The *current logical description* is the result of this second stage.

3) The third stage is where the user requirements identified during the first two stages are incorporated into the current logical description of the system. At this stage, the entity analyses and access path analyses are refined and normalisation of the data structure is undertaken. The proposed logical description is produced. This is the endpoint of systems analysis, i.e. the *specification of requirements.*
4) In the fourth stage, the logical description is inspected and various physical implementation options are determined. This is usually at different levels of automation and high levels may be expected to be more costly (with potential greater benefits) and to take longer to implement than low levels.
   - ❑ Two, three or more options may emerge, and user comments are sought on the relative importance of different desired features, not all of which may be implemented in any given option. There are also hardware and software considerations, which may restrict the range of feasible options.
   - ❑ Each of the options should be subjected to a cost–benefit analysis, either by the systems team or by management on the basis of facts provided by the systems team.
   - ❑ The options, supported by cost–benefit analyses, are the results of this stage.
5) The fifth stage is to select an option. This often involves the systems analyst in the presentation of alternatives and the possibility of recommending a single option. This naturally requires care and should be discussed with the users before any presentation.
6) After a single option is chosen, the analyst then proceeds to produce the s*tructured specification*, which is a fully detailed model of the system to be developed. Some form of data modelling will have taken place before the completion of this sixth and final stage.

4.3 The role of the systems analyst

The analyst acts as liaison between the various disciplines and levels of responsibility, ensuring that communication takes place effectively between them. This is often a difficult task. The role includes:

- ❑ communication with both users and management. Clear communication between both parties is essential if systems are to be produced which contribute to the profitability of the organisation;

- ❏ generation of the structured specification document, the production and approval of which indicates the successful completion of the structured analysis phase;
- ❏ estimation of system costs and budgets which enable system alternatives to be compared and the optimal solution selected.

Users may be hostile to the idea of change, frightened by it or just too busy trying to cope with day-to-day work to spend much time talking to the analyst. The analyst is often seen as the agent of change. The analyst must overcome user fears involving them during the course of the development as much as is practical. By doing so, not only is it more likely that the delivered system will do what users want it to, but users will buy into the project and take ownership of the system. They will then do all that they can to make the implementation successful.

The analyst must also understand the problems faced by the designer and programmers in implementing the requirements of the users, and appreciate the precise information they need to do their job.

Additionally, the analyst must resolve conflicting user requirements. Although the analyst usually has no executive authority to resolve conflicts, he/she can prioritise requirements in terms of business objectives to facilitate the appropriate decision by management.

In dealing with the different groups of people the analyst must have many skills. The analyst must have good communication skills and analytical skills. There are tools and techniques which will greatly help the analyst to do a better job. Structured techniques provide the analyst with methods of modelling (with diagrams) the existing and the new system in ways which highlight omissions and inconsistencies. The diagrams also provide a means of communication with both users and IT staff which are less ambiguous than narrative.

4.3.1 Personal characteristics of the systems analyst

With all the tools and knowledge, however, the analyst still needs certain personal characteristics to do what is a rewarding but often difficult job.

A level of analytical ability, intelligence and common sense enable the analyst to cope with the sheer volume of information which must be gathered, sifted and interpreted.

A personality which inspires confidence, radiates enthusiasm and encourages trust will make it more likely that the analyst will be given, freely and fully, the information required.

An ability to plan for the term of a project and the determination to stick to the plans, meeting all problems which arise and tackling them, is needed for the timely completion of the project. This requires stamina, resourcefulness and often a dogged persistence to stick to the objectives, in spite of difficulties, and see a project through.

4.4 Some current development methods

Here we shall examine, very much in overview, some of the best examples of well-defined methods for system development:

- ❑ SSADM 4+ version4.3;
- ❑ Object Orientation (eg UML);
- ❑ Rapid Application Development (eg DSDM).

4.4.1 SSADM 4+ version 4.3

SSADM is the most widely used method for information systems analysis and design in the UK. It was developed for Government by the CCTA in the early 1980s. With the launch of version 3 and the publication of the SSADM manual in 1986, it became publicly available and has been widely adopted by the private sector, becoming a *de facto* standard. Version 4 was launched in 1990 and has since been upgraded to version 4.3.

SSADM uses three viewpoints (Figures 4.2 and 4.3) when analysing and developing systems. This gives the developer a clear understanding of the business requirements and helps to remove errors. These viewpoints or models are based on:

- ❑ functions;
- ❑ events;
- ❑ data.

Functions represent the user's view of the processing taking place in response to an event. The event triggers the process or function.

An **event** can be anything that has an effect on the business system being considered. This may be physical, such as the receipt of a document, or time based, such as that required by the system in order to produce an end-of-day report.

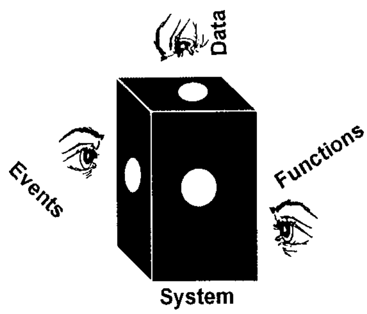


Figure 4.2 Viewpoints of SSADM.

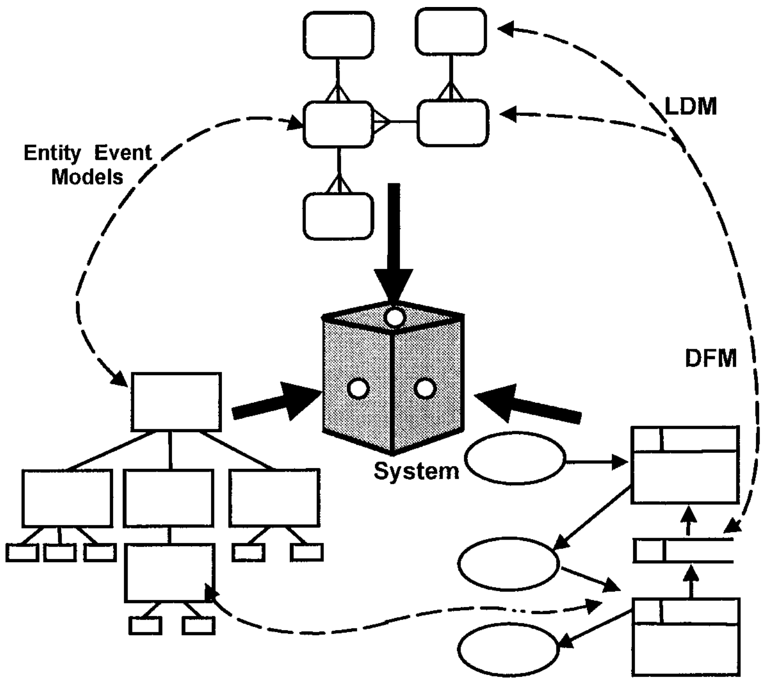


Figure 4.3 SSADM viewpoints when analysing and developing systems.

Data is the raw material of any processing system. It is the manipulation of data that provides business information. The way in which the data is structured within a system has a significant effect on the performance of that system.

The *functions* view is shown using data flow modelling. The *events* view uses entity and event modelling through entity life histories and effect correspondence diagrams. The *data* view is represented by the logical data model. (See later chapters of this book.)

The subsequent cross-checking of models from these views gives SSADM its capacity to eliminate errors, both in the perception of user requirements and in the detail of requirements specification. Figure 4.4 shows the position of SSADM within the system life cycle.

Figure 4.5 shows that SSADM v4 has four modules, each of which has an interface with the activity of project management and control. Progress can be measured in terms of modules completed, determined by the production of a set of specified products from each SSADM module. Similarly, products from one module act as input to other modules. Each *module* consists of one

or more *stages* which contain a number of *steps*. For example:
The requirements analysis module
Stage 1: Investigation of current environment
  Step 115: Develop business activity model
    Task ....
  Step 120: Investigate and define requirements
    Task ....
  Step 140: Investigate current data

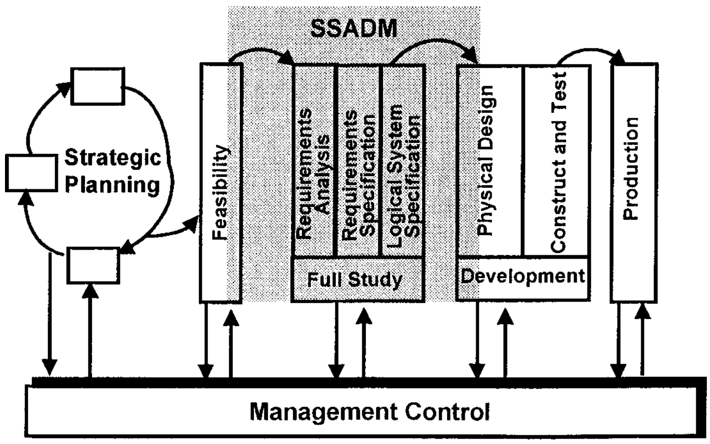


Figure 4.4 Position of SSADM within the system life cycle.

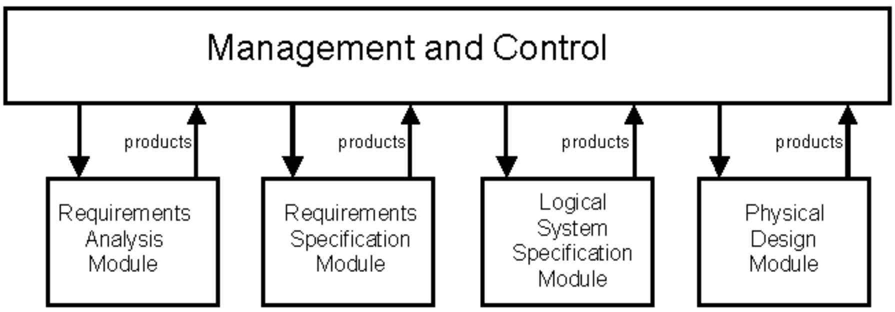


Figure 4.5 Four modules of SSADM.

4.4.2 SSADM module description

RA module

Requirements Analysis (RA) is concerned with information gathering in order to understand fully the business area which is the subject of the proposed system. It also identifies 'owners' of the system. Requirements are documented in the *requirements catalogue* and expressed in terms of functional and non-functional requirements. Functional requirements are *what* the system should do, whereas non-functional requirements describe how, how well, or to what level of quality a facility of the system should be provided, e.g. security, service level requirements or usability.

This information feeds into the process of defining and selecting business systems options (proposed solutions).

RS module

The objective of the Requirements Specification (RS) module is to take the agreed business system option and develop it in order to add precise details of data, functions and events to be present in the future system. The output requirements specification is of such detail as to allow development of the system to proceed, whether this is carried out in-house or contracted out to a software supplier.

LSS module

The Logical System Specification (LSS) consists of two stages: Technical Systems Options (TSO) and Logical Design (LD). The TSO establishes possible technical environments within which the system can be physically developed and operated. It is concerned with *how* the system will be implemented.

Logical design's first step is to define user dialogues. It then covers the further definition of functions in the system.

PD module

The objective of the Physical Design (PD) module is to specify the physical data processes, inputs and outputs, using the language and features of the chosen physical environment and incorporating installation standards.

4.4.3 Object-oriented modelling

There are well in excess of 50 object-oriented methods currently in use. The leading ones include the methods evolved by Rumbaugh, Booch, Shlaer and Mellor, Coad and Yourdon, and Jacobson. About a dozen methods dominate the field, whereas the rest only achieve marginal interest by practitioners.

Rumbaugh's Object Modelling Technique (OMT) is touched upon briefly here with figures showing examples of the three viewpoints that OMT takes of a system:

- ❑ object model;
- ❑ dynamic model;
- ❑ functional model.

The *object model* (Figure 4.6) describes the classes and relationships within a system.

Objects know things about themselves and can provide information when operations are performed on them. An object interacts with other objects outwardly through its operations. For example, the clock can, on request, display the time. The data, or attributes, that belong to an object can only be accessed via a message causing one of its operations to execute, but the data is concealed from other objects.

An object *class* defines a group of objects with similar properties, behaviour and interactions. The class CLOCK can define an alarm clock, a grandfather clock, a cuckoo clock, etc. The alarm_clock, for example, *inherits* the properties of CLOCK, e.g. Display_time, Keep_time, Set_time, as well as having its own properties, e.g. Display_wakeup_time.

The *dynamic model* (Figure 4.7) describes *when* things happen within the system.

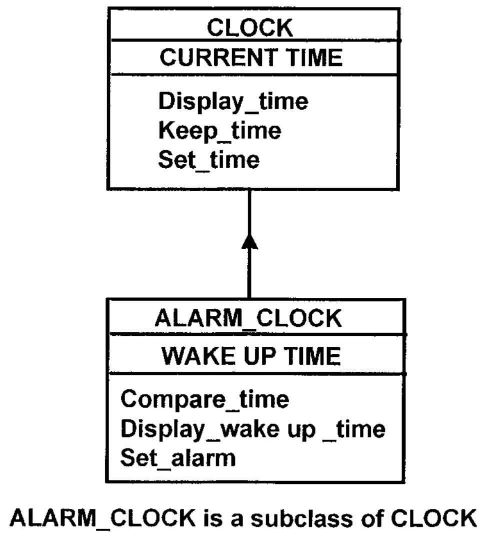


Figure 4.6 The object model.

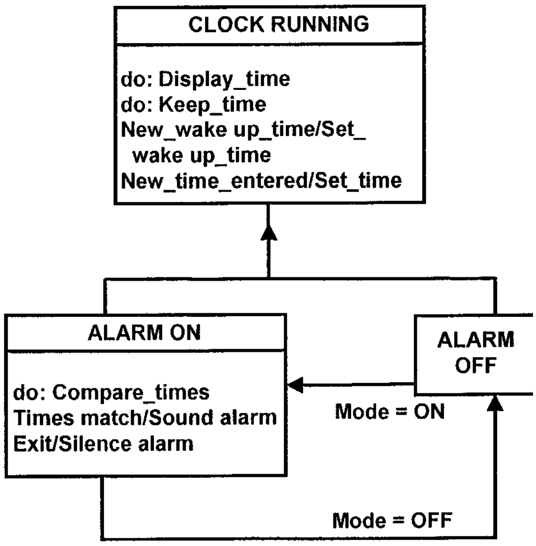


Figure 4.7 The dynamic model.

The *event* of switching the alarm on (Mode = ON) causes the actions 'Compare_times', if 'Times match' then 'Sound alarm' and 'Exit/Silence alarm'.

The *functional model* (Figure 4.8) describes *what* happens within the system. The parallel lines contain the information and the ovals describe the processes which change the data.

Before leaving this overview of object-orientation it is important to mention the development of a new standard in O-O Modelling, the Unified Modelling Language.

4.4.4 Unified Modelling Language (UML)

In 1997 Booch Rumbaugh and Jacobson pooled their ideas and formulated the Unified Modelling Language (UML) which has become widely accepted as a standard.

UML uses a number of diagrams to describe the system under development:

❑ Static Structure Diagrams
- Class diagrams
  Shows the static structure of the system;
- Object diagrams
  Shows one instance of a class diagram;

❑ Use Case Diagrams
Show typical interactions between a user and the computer system;

❑ Sequence Diagrams
Shows an explicit time sequence of messages passed between objects;
❑ Collaboration Diagrams
Shows the relationship between objects and their links, used when sequence in time is not as important;
❑ State Diagrams
Shows all the possible states that an object can exist in and the events that bring about the changes in state;
❑ Activity Diagrams
A special case of a state diagram showing the sequential flow of activities and the decision points which influence the direction of workflow within the system;
❑ Implemementation Diagrams
- Component diagrams
  Show the structure of the code – the software components that will be used to build the system and the relationships between them;
- Deployment diagrams
  Shows the structure of the runtime system.

There are many texts devoted to UML and the reader is recommended to refer to these for a detailed discussion.

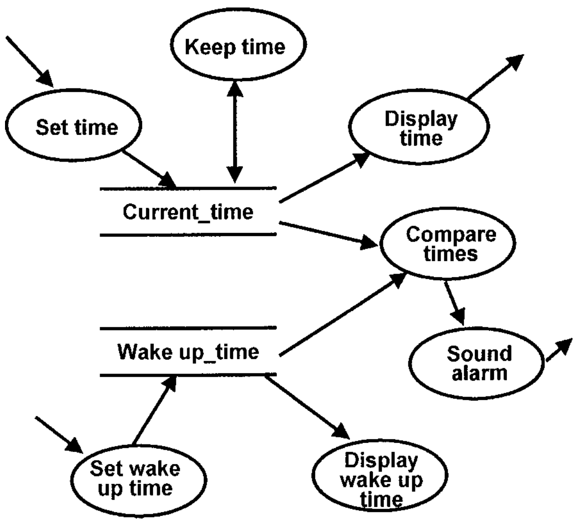


Figure 4.8 The functional model.

4.4.5 Rapid Application Development

Several established methods for systems development have sought to define an approach which is appropriate for Rapid Application Development. In the mid nineties a consortium of major companies was formed in the UK to pull together the best techniques for RAD into a coherent framework. By incorporating the ideas of James Martin, Dupont and others together with the experience of those within the consortium the Dynamic Systems Development Method (DSDM) was formulated and has subsequently been refined.

The DSDM life cycle is shown in Figure 4.9.

The life cycle has five phases:

❑ Feasibility Study;
❑ Business Study;
❑ Functional Model Iteration;
❑ Design and Build Iteration;
❑ Implementation.

The development life cycle diagram shows an iterative approach moving anticlockwise from the top. The arrows show the transfer points from one phase of the life cycle into the next and where the development can easily return to an earlier phase.

Each of the development phases (the functional model and the design and build iterations) contains iterations through prototyping. Basic controls must

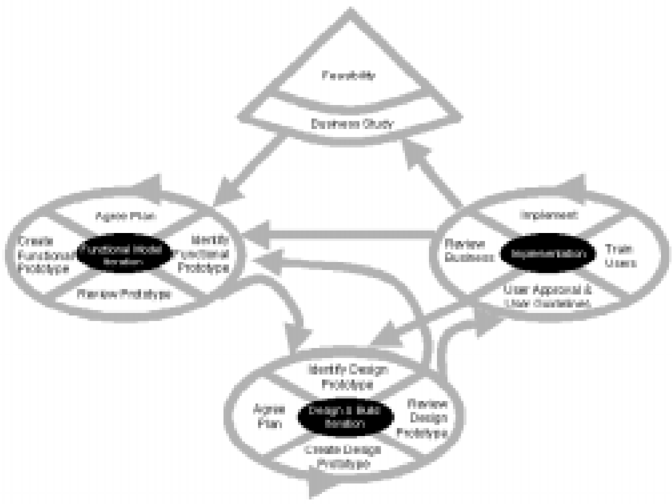


Figure 4.9 DSDM life cycle.

be implemented in order to assure the success of these activities. These controls are built into a prototyping cycle, which consists of four stages:

- ❑ identify prototypes;
- ❑ agree schedule;
- ❑ create prototype;
- ❑ review prototype.

As a general rule, it is recommended that three iterations are allowed for a phase (functional, design and build):

- ❑ initial investigative prototype;
- ❑ refining prototype;
- ❑ consolidating prototype.

The method addresses the following within project development:

- ❑ Project Management;
- ❑ Team Structures;
- ❑ User Involvement;
- ❑ Prototype Management;
- ❑ Skills and Responsibilities;
- ❑ Estimating;
- ❑ Risk Assessment;
- ❑ Change Control;
- ❑ Configuration Management;
- ❑ Development Environment;
- ❑ Method Tailoring;
- ❑ Testing;
- ❑ Quality Assurance;
- ❑ Software Procurement.

DSDM is based on **nine principles**. Each one of the principles is applied as appropriate in the various parts of the method:

- ❑ active user involvement in rapid application development is imperative;
- ❑ DSDM teams must be empowered to make decisions;
- ❑ the focus is on frequent delivery of products;
- ❑ fitness for business purpose is the essential criterion for acceptance deliverables;
- ❑ iterative and incremental development is necessary to converge on an accurate business solution;
- ❑ all changes during development are reversible;
- ❑ requirements are baselined at a high level;
- ❑ testing is integrated throughout the life cycle;
- ❑ a collaborative and co-operative approach between all stakeholders is essential.

DSDM provides a framework for building and maintaining systems that meet tight time constraints through the use of various techniques and the previous nine guiding principles. The following are the major techniques used within DSDM:

❑ Facilitated Workshops;
❑ Prioritisation of Requirements (**M**o**SC**o**W**) – requirements are categorised as:
  - **M**ust have [the minimum usable set]
  - **S**hould have [could manage without if absolutely have to]
  - **C**ould have [if development allows include these next]
  - **W**on't have [want to have but won't have during this development);
❑ Timeboxing (a sequence of short timeframes within each of which part of the development is scheduled to take place);
❑ Prototyping.

A DSDM project is able to deliver on time because the delivery date is absolutely fixed whereas exactly what is delivered (the functionality) is not. The reader should note that the project will deliver a quality product as that feature of the product, like delivery date, is not open to negotiation. Timeboxing monitors the delivery of the project in incremental parts and if the development is behind schedule the implications are that some of the lower priority requirements will be dropped from the project.

The discussion here is simply an overview of DSDM and the reader is encouraged to read more in depth treatments of RAD and DSDM which are listed in the Bibliography at the end of this book.

4.5 Summary

In this chapter the background to the need for a more disciplined approach to systems development has been discussed, together with the costs of and an approach to introducing a method. Our discussion then took a generic look at what a method should do during the analysis phase of the system life cycle and also what the systems analyst was required to do. The skills which the analyst needs to accomplish the role were also discussed.

Finally, three specific methods (SSADM V4+, O-O Modelling and DSDM) were discussed in overview.

We have now set the context for systems development in terms of:

❑ business organisation;
❑ the nature of information;
❑ systems and the systems life cycle.

The rest of this book will focus on the analysis techniques which form the tools available to the analyst to aid the process of system development.

Exercises

4.1 You have just joined a company and have been asked to look at the way in which software is being developed and to suggest ways of improving the process. Your preliminary investigations reveal that the user department requests a system to be developed and, if the IT department has available staff, the programmers go away, write the programs and hand it over to the users when complete. Write down your preliminary thoughts and suggestions of how the situation may be improved including any costs and benefits to be considered.

4.2 What is the System Analysts's role and what skills are needed by someone in this role?

4.3 Describe the views of the system that are taken by SSADM and name the techniques used to model these views. What is the major advantage of these models of the different views?

4.4 Name the diagrams used in Unified Modelling Language (UML) and state what they describe.

4.5 What are the Nine principles of DSDM and what are the four major techniques used in the approach?

5
Fact-finding

Introduction

This chapter looks at *investigation*, an activity which is the first step of analysis and proceeds through all analysis stages. The term *fact-finding* is also used for this aspect of the analyst's job. Information gathered during the investigation will concern not only the existing system but also what the user might require from a new one. The analyst must recognise these two aspects and document them separately.The chapter deals first with some of the general problems associated with the investigation activity under the following headings:

- ❑ thorough investigation;
- ❑ business activity modelling;
- ❑ dealing with the user;
- ❑ changes;
- ❑ the specification.

After a summary of the problems, we re-visit the system life cycle, and look at:

- ❑ terms of reference;
- ❑ methods of investigation;
- ❑ user Analysis;
- ❑ procedures, data and documentation.

5.1 Thorough investigation

Thoroughness, alongside investigation of the right aspects of a system, is the key to a successful investigation. Frequently, an analyst has only partly

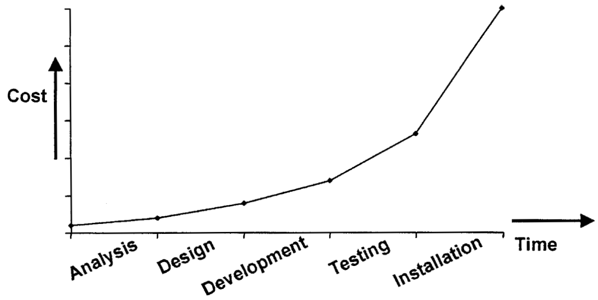


Figure 5.1 Cost of change related to development stage of new system.

understood what an existing system does and what a new system needs to do. As a direct result, a new system, however closely it matches the analyst's description of the requirements, fails to match the organisation's requirements. The wider, more obvious requirements are met, but it is in the detail that the new system fails. Perhaps it does too much, perhaps it does too little. In either case it needs to be modified. Modifications to systems are not uncommon. Some may be required because of changes in policy or the environment in which the systems operate, whereas other modifications are required because of errors in the system due to inadequate analysis or design. Barry Boehm was one of the first analysts to publish a chart showing the cost of a change related to the stage that the development of a new system has reached. He showed that:

- ❑ the earlier an error is discovered and the system altered, the cheaper the alteration will be;
- ❑ an error discovered during testing and installation will cost much more to correct than the same error found early on.

This is illustrated in Figure 5.1.

Time and effort devoted to investigation and analysis, to ensure that the information collected is accurate, factual and complete, is time and effort well spent.

The 'right things' for the analyst to investigate, for both current practice and requirements are:

- ❑ the processes/procedures (how);
- ❑ the data (what);
- ❑ the people involved (who);
- ❑ the reasons why things are done (why);
- ❑ where they happen (where);
- ❑ when they happen (when).

If the analyst keeps in mind to ask questions beginning *what, why, where, when, how, who*, these aspects will be elicited.

5.2 Business activity modelling

5.2.1 Introduction

When investigating the Information Technology (IT) requirements for a new system, the definition of requirements is of better quality if it is based on the wider knowledge of the business environment within which the users operate. The definition of requirements is improved further if the analyst understands not only what the users do but why they do it and how different users' business activities are interrelated.

The Business Activity Model (BAM) is a means of documenting and analysing those activities which are essential for the business to meet specific objectives. It enables the analyst to understand the overall business environment and develop requirements to ensure that the new computer system will support the attainment of the business objectives. The BAM is not concerned with who carries out the activities or where they are carried out; these aspects are covered later in Work Practice Modelling when the BAM is mapped into the organisation structure.

In practice, the IT system may only provide part of the information support required by the business activities, but the BAM technique will help the analyst and user to identify the full information support requirements (i.e. manual as well as computerised) and ensure that the new system interfaces smoothly with the user's overall job.

There are several well-established approaches to modelling business activity and each organisation needs to decide which is the most appropriate. Two forms of notation are used here as examples.

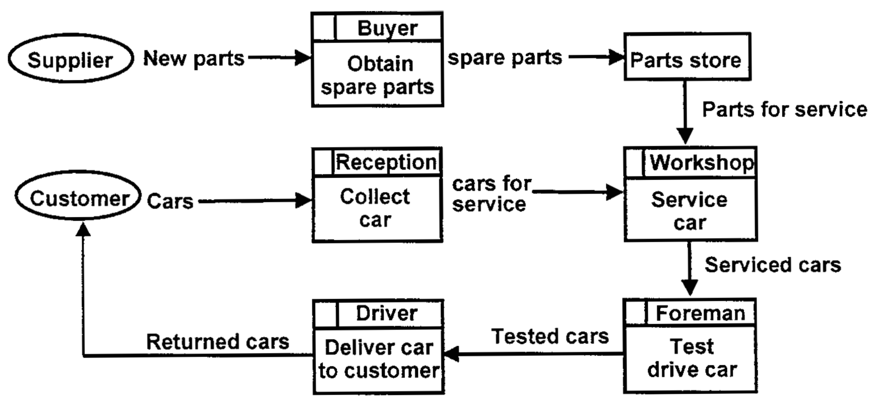


Figure 5.2 Example of notation for modelling business activity.

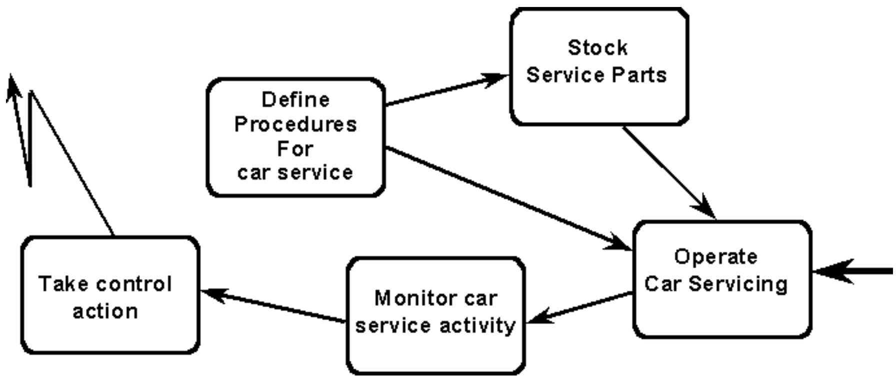


Figure 5.3 A notation for business activity modelling.

5.2.2 BAM components

The BAM should contain four components:

❑ the business perspectives, stating *why* the business does what it does;
❑ the business activities and their interactions, describing *what* is done;
❑ the business events, indicating *when* business activities are carried out;
❑ the business rules, defining *how* activities should be carried out.

Business perspectives. All businesses should be able to encapsulate in words a statement of what everyone believes the basic function of the business to be and what it is achieving (i.e. why the business is in existence). Such a statement is called a *business perspective*.

For example, a company which sells and services cars, may be described from the following perspectives:

❑ A system to produce an acceptable return on investment by servicing cars;
❑ A system to maintain loyalty of existing customers by providing a high-quality service;
❑ A system to produce acceptable return on investment by selling new and used cars;
❑ A system to retain customers by providing a lease/buyback scheme.

Stating and understanding the business perspectives will help to determine what has to be done for the business to be successful. Measures of performance and control action required to keep the business on track will all be based on the business perspectives.

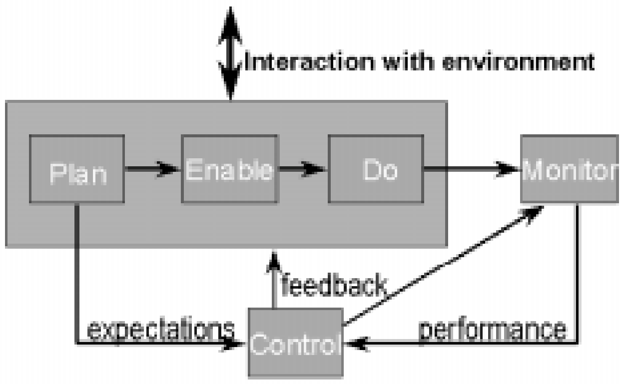


Figure 5.4 Five different types of activity are represented in a BAM.

Business activities. A *business activity* is an activity directed at some specific purpose, such as selling cars or servicing, referred to as the *primary task*. A logical activity model documents the essential activities to achieve the primary task.

Five different types of activity are represented in a BAM:

PLAN, ENABLE, DO, MONITOR, CONTROL

The model must include doing and enabling activities. Doing activities are essential parts of the primary task and contribute directly to the business purpose. Enabling activities ensure that the resources and facilities needed by the doing activities are available; these would include recruiting and training. Doing and enabling activities must be planned. Part of the planning activities defines rules dictating how activities are done and sets performance expectations. These involve setting targets for sales, establishing times for carrying out services, etc.

Planning enabling and doing activities needs monitoring so that performance data can be compared with expectations. Control activities act on the other activities when performance expectations are not being met.
In many IT projects the major requirements are concerned with building cost-effective monitoring to provide improved support for the planning and control activities.

Business events. Activities in the model are triggered by *business events*. The approach selected for business activity modelling must have a means of documenting business events and relating them to the activities that they trigger.

There are three types of business event:

- ❑ external inputs from outside the system boundary;
- ❑ decisions made within the business activities (e.g. price changes
- ❑ scheduled points in time. These might include the start of the day, end of the day, quarterly etc.

Some business events may trigger more than one activity. In addition, some activities may be triggered by more than one business event. Although it should be possible to relate the information system events used in entity life histories back to the business events, there is not necessarily a one-to-one correspondence. Some business events may not be handled by the computer system if it is decided that the particular activities they trigger are beyond its scope.

Business rules. For many activities identified in the BAM there are rules defining how the activities are carried out. Wherever rules exist they should be referenced from the business activities.

Rules are of two types:

- ❑ constraints: these describe conditions under which the activity *is* or *is not* carried out;
- ❑ operational guidance: this determines *how* activities are carried out. It may be defined procedurally or may refer to external sources (manuals, legislation, etc.).

5.2.3 Information support for business activities

Many business activities need information support, some of which may be provided by a computerised information system. The supporting information system should cover all stored information needed by the business activities and may contain non-automated processes as well as processes supported by computerised information systems.

An information system can be used to support business activities in two ways:

- ❑ by automating the business activity (or parts of the activity), e.g. the calculation and production of invoices could be computerised;
- ❑ by providing information to the business activities, e.g. the selling activities of a company could be supported by a computer-produced report listing the monthly sales of each member of the sales staff.

5.2.4 The BAM product

There are several well-established methods for the development of BAMs and there is a range of notation for representing them. One approach is to use resource flow diagrams (Figure 5.2), which are part of the data flow modelling technique.

The resource flow diagram is mainly concerned with *doing* and *enabling* activities. Planning, monitoring and control activities are not usually shown. However, it is possible to identify the relevant information support and the business activities requiring the support by asking appropriate questions in relation to the resource flow diagram. For example, planning activities can be identified by asking the following questions.

- what has to be done to decide what categories of resource should be offered?
- what has to be done to stay competitive?
- what has to be done to monitor the consumption of the resource?
- what has to be done to set performance expectations?

Having identified the business activities, it is possible to decide whether information is needed by that activity, and if so the source of the information can be determined, e.g. from an external source, from a non-IT internal information source or from an IT database.

Clearly defined requirements for the new system can be determined from this kind of business modelling and critical analysis.

5.2.5 Formal systems model

Regardless of the approach or notation used to document the BAM, the resulting model may be validated by comparing it against a formal model of a 'human activity system'. This states that the following mandatory elements should exist or be defined:

- *objectives and purpose:* these must be explicit;
- *connectivity:* the activities in the model must all be connected;
- *measures of performance:* must exist and expected levels of performance must be set;
- *monitoring and control mechanisms:* performance data must be collected and compared with expectations. There must be control activities which have the power to change other activities when expectations are not met;
- *decision taking procedures:* there must be decision-taking procedures which will be influenced by the control actions;
- *boundary:* the extent of the system must be clear and communications across the boundary explicitly defined;
- *resources:* staff, materials and other resources used by the system must be acquired, allocated, replenished and accounted for;
- *systems hierarchy:* it should be possible to decompose the system hierarchically based on the scope of control activity. Any business activity

should be within the scope of only one control activity. If not, additional activities need to be introduced to resolve any conflict.

5.2.6 Requirements definition

The BAM should be the main driving force behind the requirements definition. Requirements may be:

- ❑ Functional: those features or facilities needed to satisfy the information needs of the user. Basically, they cover *what* the system should do;
- ❑ Non-functional: these describe *how well, or to what level the system should work.* Non-functional requirements include:
  - ❑ service levels;
  - ❑ usability;
  - ❑ access restrictions and security;
  - ❑ monitoring;
  - ❑ audit and control;
  - ❑ conversion from the current system;
  - ❑ interfaces with other systems;
  - ❑ archiving.

Functional requirements can be derived from the BAM by considering the following:

- ❑ operational activities may need information support;
- ❑ some business activities may be candidates for automation;
- ❑ monitoring activities may need information to be collected;
- ❑ activities that resolve conflicts and apply control actions may need enquiry and browsing facilities or some form of decision support;
- ❑ business activities which maintain data within the database require data entry, validation and update functionality.

Some non-functional requirements may also be derived from the measures of performance specified in the BAM.

5.2.7 Summary of BAM

The definitions of requirements will be of better quality if the analyst has a sound understanding of the business activities that the new information system is to support.

Business activity modelling is a technique that can assist an analyst's understanding by focusing attention on:

- ❑ *why*: the business perspective;
- ❑ *what*: the logical activity model;
- ❑ *when*: the business events;

❑ *how*: the business rules.

The BAM is the major input to work practice modelling, described later in this chapter as part of user analysis. The Work Practice Model (WPM) is the mapping of the business activities on to the user organisation. In particular, it specifies:

❑ the actors (user roles) who carry out each business activity;
❑ where the activities are carried out.

5.3 Dealing with the user

The analyst needs to keep in mind the fact that users, at all levels, have a job to do. The amount of time that they can usually allocate to describing their work, checking that the analyst has correctly understood it and generally working with the analyst is likely to be very limited. However, recent user-centred approaches to systems development require appropriate users to be allocated to the development team for considerable proportion of their time (possibly even full time for a short period). This can be very effective, provided that the user does not also have the whole of their usual job to perform at the same time. Whether the user's time is in short supply or not, problems of communication still arise, as described below.

5.3.1 Missing the obvious

A user knows their function in the existing system, and often considers it to be obvious. However, it may not be obvious to the analyst, who will need care and tact to obtain the relevant information. The difficulty is in trying, in a very short time, to understand what is done and why. The people who operate the system are completely familiar with it, whereas the analyst is not. They know what is important and what is merely desirable.

The analyst should try to identify the important parts of the policy being implemented. He or she cannot expect to receive a clear, detailed and accurate account of precisely what is done, why and with what priorities. The analyst must help the users to provide sufficient information to enable such a description to be produced and must check it with them to ensure that it is right. Diagrammatic modelling techniques can be very helpful in identifying omissions and inconsistencies in the information gathered.

5.3.2 Instant design

While the analyst is talking to people, listening and recording what they say, his or her mind will be busy 'pigeon-holing' the information. The analyst will also, quite unintentionally perhaps, be envisaging how a computer system could be used, or the existing computer system modified. The problem at this stage is that the analyst will often be asked questions about the effect that a particular approach to automation will have.

The analyst should take great care at this early stage. He or she should not commit anyone to a particular path, and should make no promises or suggestions. The people to whom an analyst talks may see their jobs threatened, or have unrealistic expectations raised regarding the system they will eventually receive. The analyst does not usually have the authority to make such decisions and at this stage no definite decisions have been taken regarding future developments.

Off-the-cuff 'instant design' comments should be avoided. Apart from any effect they may have on the people who hear them, the comments might be wrong.

5.3.3 User comprehension

It was noted earlier that the analyst must check the results of the investigation with the user. This is easily said, but it carries a series of implications.

Firstly, the user must be able to grasp what has been presented to them quickly enough to be able to consider it, and the analyst must be able to discuss the details and make notes of any corrections.

Secondly, the user is not one person, but typically three different groups of people within a hierarchy. These are the senior management, responsible for policy; the line management, responsible for implementation of that policy and the individual users, who perform the day-to-day work. Each has an important contribution to make and each works at a different level of detail. Therefore the analyst must ask the right questions at the right level and also present the written and diagrammatic results at the right level. Often the senior level wants to see a logical model of the system showing what the policies are. The other levels may want a physical model showing how the policies are carried out.

5.3.4 Office/factory politics

Developing a production control system, which includes inventory control, provides an example of the problems that politics can present. If the inventory controller is a forceful individual, it may be found that although senior management sets the company's prime policy of meeting sales targets and satisfying orders, the analyst finds him or herself to be investigating a system in which the stores function actually controls the organisation. Further, if part way through the development of such a system, the inventory controller is promoted to production manager, then the detailed system requirements may change overnight. The problem of change is looked at later in this chapter.

There is usually very little that can be done about internal politics, the way authority works and the influences these can have. The nature of an analyst's work is such that any new system might have a political effect by shifting authority or changing the distribution of power. However, analysts can be helped at the outset if the boundaries of their work are made clear, and the objective of the system which they are required to produce is made equally clear, is precisely known and is understood by the users. In this way, the analyst's work is less threatening to a user at any level and is therefore less likely to create political upsets.

However, systems analysis is working with people, and for people everywhere 'survival' is the driving force behind their decisions and actions. Therefore, political influences will not go away. They will be less of an interruption and they will affect the analyst less if he or she is seen to be working to a defined brief given by senior management.

5.4 Change

One of the worst features of installing a computer-based system in any part of an organisation is that the procedures followed by that part of the organisation are likely to change within the duration of the project.

It is all very well to talk about 'freezing a specification', but the result of doing this is to guarantee that the system developed from it will be 'yesterday's system' and not 'today's', and will not fit the business need when it is delivered.

Far more useful and relevant to the changing needs of business, is a system which is specified right from the start in such a way that when change is needed, as it will be, it can be introduced into the records of the analyst, carried through into the design, and implemented without the need for a massive programme of modification. No system is immune from change, so a good system starts life with a thorough and accurate investigation, the results of which must be noted down clearly, unambiguously and as briefly as possible. This document, the results of the investigation and the other phases of analyses, comprises the next, and last, major problem which the analyst must overcome.

5.5 The specification

It may be asked why the specification has been included in this list of 'problems of analysis'. The reason is that however well the investigation is carried out, however accurately it has established what any new system needs to do and why, a new system is only appropriate if the specification is comprehensible to both the users and the designers.

Historically, for want of a better method, specifications were produced in narrative form, with some charts and forms to specify detail. Narrative is inherently vague and long-winded. Trying to remove the vagueness by careful choice and definition of words makes the narrative even longer and less comprehensible. Charts and special-purpose forms are normally created for computer people, not users. The result is that the user is presented with so much detailed material about the system that it is almost impossible to read, absorb and understand. The user does not like to admit this, so he or she agrees the specification, hoping that it is right. It then passes to the designer, who may be able to understand it better than the user, but it is still a huge document. This is often ambiguous, so even if the specification is correct, it is quite likely that the designer will misunderstand some of it.

This is the problem of the specification. It is too 'wordy', too imprecise, often incomprehensible to the user, and an unsuitable way to present the detailed requirements.

5.6 Summary of the problems

The problems discussed are of two types. There are those for which methodologies have been developed to reduce their impact, and others which rely on the interpersonal skills and professional integrity of the analyst. In this latter category are the need for thorough investigation and the production of documentation. However, documentation can be assisted by methodology. The avoidance of 'instant design' and office or factory politics also fall into this category. If a thorough and accurate investigation is not carried out, there is a high probability that the resulting system will be inferior to one that has been fully investigated. The analyst must not be led astray by the idea that the original system being investigated does not matter because it will all be changed anyway. The reason for documenting the existing system is to ensure that the new system preserves the important policies and rules contained therein.

Naturally, as much care as possible must be taken to collect information, without influencing the people who are supplying the facts. They are people, not machines, and they may have been in the organisation for a long time. All offices and factories have the potential to become a political battleground. A system change almost always changes the power structures of an organisation, and therefore great care is needed in its introduction.

The other problem areas can be greatly influenced if there is a method for recording and analysing the details collected, and which is also comprehensible and self-checking. The techniques described in this, and other, chapters are designed to meet this requirement.

5.7 The system life cycle

It would be very easy to suggest that all investigations are simple, clear-cut,

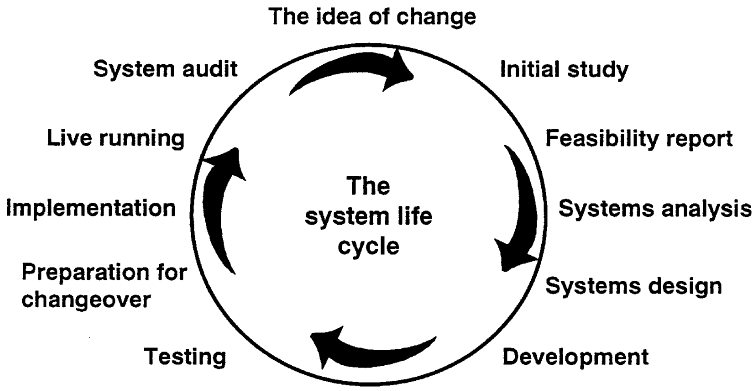


Figure 5.5 The system life cycle.

well defined in advance and are one step in a defined sequence. Unfortunately, that is seldom true.

There are, at all the stages of a system, endless possibilities of back-tracking, of finding something in one stage that invalidates the work done at a previous stage. Therefore, valid though the concept is, one must be aware that it is an over-simplification. Also, and this may be somewhat less obvious, the environment in which the system lives alters with time, and is altered by the presence of the system itself. The change may be something straightforward, such as a new product to be manufactured, a reorganisation, or some new policy towards customers. Alternatively, the alteration may come from the users who are stimulated into thinking about what other facilities their computer systems could provide. Consequently, they may see, and require, some new facility they did not previously identify.

Any new requirement cannot simply be introduced into the system cycle at a random point. It must be incorporated into the policies expressed at the logical system definition, and then carried through to catch up with the point reached by the system development.

The system life cycle (Figure 5.5) consists of a whole series of feedback loops, which in total should result in progress. It may also contain several forward cycles, to process changes, the later ones being processed forward as quickly as possible, to catch up with the main cycle. If the area to be investigated already contains a computer system it should be kept in mind that the existing system may not be appropriate for the current situation.

5.8 Terms of reference

Before an analyst can start the investigation he or she requires a statement of what needs to be done. This statement is called the *terms of reference*. It sets out the:

- ❏ objective of the investigation;
- ❏ boundaries;
- ❏ allowable expenditure and other resources available;
- ❏ time by which it should be completed;
- ❏ method of presenting the results.

These terms should come from senior management, which will give the analyst credibility and authority. As can be seen from Figure 5.5, investigation, which is part of systems analysis, is not the first phase in the system life cycle (although it is the first stage in analysis). An earlier feasibility study and report will normally have been completed. As a result of that study the decision has been made to undertake a detailed investigation. Also from the initial study, the terms of reference for the detailed investigation can be established. It must be noted that the decision made as a result of the initial study is to investigate and not to implement.

5.8.1 Objectives of the investigation

The initial study is a miniature investigation of its own, which highlights areas that need a more detailed investigation, and also provides rough costs and benefits expected if a new system were implemented. This information can be dangerous and misleading during a detailed investigation. The analyst must not think of solutions until the work of analysis is complete. However, the feasibility report spells out the basic objectives that should be met by a new system and this helps the analyst to determine user requirements.

The objectives of the investigation should be spelt out in terms of procedures or functions to be investigated and problems to be resolved. They state the management requirements from the system.

5.8.2 Boundaries

The boundaries of an investigation are usually stated as departments, groups or functions that are to be studied or sometimes excluded from the study. If the analyst can be given a list of those people or groups who provide input to the area to be investigated, and a list of those who receive output from it, i.e. the *sources* and *sinks*, then he or she has, by implication, been given the boundaries. (see Figure 5.6).

The people or department that are the sources of input data and those that receive the output, the sinks, are outside the area to be investigated. Everyone and everything in between are in the *domain of study*. However, this statement assumes that the analyst can identify quite precisely what data flows into

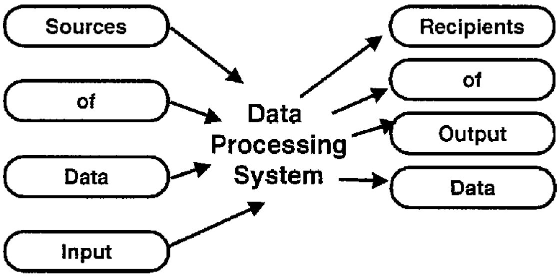


Figure 5.6 Sources and sinks defining the boundaries of an investigation.

and out of the domain of study. Sometimes this is not true and, as the investigation progresses and problems and requirements are identified, what initially appeared to be a source or a sink may need to be brought within the system boundary and investigated.

Occasionally, a source or sink may be outside the organisation being studied, and thus not be available for investigation, e.g. customers or suppliers. Where the source or sink is inside the organisation, it may be appropriate to include it in the investigation to ensure that all requirements are met.

5.8.3 Costs and timescales

Cost and time are normally two sides of the same coin. Cost is the number of person-days at the 'going rate'. However, time may have another meaning, i.e. elapsed time.

Not all investigations are carried out by one person full-time. If a particular investigation is carried out this way, the timescale for completion will be a number of person-days and the cost will be that number multiplied by the current person-day rate. Normally there are two separate figures, one for person-days (expressed either in days or as a cost) and the other for the elapsed time or the required completion date.

5.8.4 The requirements catalogue

The analyst should build up details of the requirements for the new system throughout the investigation process. This can be done in the form of a requirements catalogue. The requirements catalogue is the central repository for information about requirements. The format will be dependent on the case tool which is being used to manage the catalogue but at least the following information should be recorded for each requirement:

- ❑ requirement ID – a unique identifier which can be used as a cross-reference on other documents which record how the requirement is implemented;
- ❑ requirement name;
- ❑ business activity – supported by this requirement;
- ❑ source – person, document, SSADM product;
- ❑ priority – e.g. high / low or mandatory / desirable / optional;
- ❑ owner – person with responsibility for negotiation about the requirement;
- ❑ functional requirement;
- ❑ non-functional requirement;
- ❑ benefits – expected from implementation of this requirement;
- ❑ comments / suggested solutions;
- ❑ related documents;
- ❑ related requirements – if different requirements influence each other or conflict they should be cross-referenced;
- ❑ resolution – this would normally be a reference to a function definition. If a decision is taken not to implement a requirement then the reason should be recorded.

5.8.5 Presentation of results

An analyst may be working alone or as a member of a team. All team members need a common method of recording the findings of the investigation, as well as a common approach to the investigation. From these records the results of the investigation will emerge.

These results, both user and IT, are normally presented to management in three ways:

- ❑ written report, including diagrammatic models;
- ❑ oral presentation;
- ❑ prototype.

All reports written by the analyst should be clear and concise. To achieve this the analyst must have clear objectives and know who the recipient(s) will be. Management will expect a report aimed at their level. It should outline what the existing system does, the problems associated with it, and the additional requirements that should be contained in any new system, in sufficient but not excessive detail.

The oral presentation supports the written document. It enables the analyst to highlight the most significant points and allow management to ask questions.

A demonstration prototype can be developed to show the analyst's understanding of the requirements and to demonstrate what is possible. It is used to extract feedback but does not always incorporate that feedback. It may be a one-off process, or may be intended as an iterative one, where the prototype will be evolved into the eventual system. The aspects to be considered before prototyping are:

- ❑ check that the appropriate tools are available for developing a prototype;
- ❑ ensure that expertise is available to develop the prototype *rapidly*;
- ❑ select the key areas to demonstrate;
- ❑ test the prototype before the demonstration;
- ❑ identify the points for further discussion and ensure that the prototype covers these;
- ❑ ensure that features shown in the prototype would actually be feasible in the eventual target environment for the system.

5.8.6 Modifying the terms of reference

It is by no means uncommon for analysts to discover that some of the terms of reference need to be modified in the light of detailed knowledge. It is therefore better to limit the initial terms of reference to a statement of boundaries and timings. Then, after the initial investigation, the terms of reference can be expanded to include new objectives and widen the scope of a full investigation.

5.9 Methods of investigation

5.9.1 Introduction

Having established the terms of reference, the analyst can start the investigation. This stage is often called the 'fact-finding' stage, and it is worth considering the word 'fact' and asking how 'facts' differ from 'data'. The British Civil Service defines 'fact' as:

A fact is a verified item of data.

The word 'verified' is important, as will be seen when looking at various methods of fact-finding. The analyst is often verifying data collected by one method against similar data collected by another method, and if the data agree then he or she probably has a fact. One of the problems of fact-finding is that of perception. One person may perceive a situation in one way whereas another person may perceive the same situation in a completely different way. The analyst's job is to obtain facts and this is often not an easy task.

The first step in fact-finding is to study background information on the area to be investigated. Business activity modelling gives the analyst a sound understanding of the overall business environment. Other useful information includes:

- ❑ how the organisation is structured: the relationships between the various divisions or departments, who holds the key responsibilities in these areas, their styles of management and their interest in the current investigation;

- ❏ previous studies that may have been carried out and their outcome: were the results accepted or turned down, and why?

Having studied the background information, the analyst is ready to proceed with the investigation. A number of methods can be used to collect facts, the analyst must choose which methods most suit the investigation and timescale. The methods are:

- ❏ observation;
- ❏ record searching;
- ❏ special-purpose records;
- ❏ sampling;
- ❏ questionnaires;
- ❏ interviewing;
- ❏ workshops (facilitated workshops).

5.9.2 Observation

Observation can be formal or informal, and both are equally useful.

Formal observation is a planned, conscious approach, which involves watching an operation or procedure for a specific period. It is best to avoid long periods of observation as it could be unnerving to those being watched and cause lack of concentration in the observer. The analyst should also be aware that the act of observing can change people's behaviour. The analyst might use formal observation to gain an understanding of a clerical or manual operation (e.g. a clerk completing a form or a manual worker carrying out a task). The technique can be used to clarify what users have already said about their tasks and can also be used to trace bottlenecks or blockages in the system.

Informal observation should be used by the analyst at all times when investigating user operation. Basically, this means being alert, but it is not as easy as might be thought. So much visual and oral information reaches the brain that it filters out all but the most important. However, when carrying out an investigation, the analyst must look and listen for every detail relevant to the investigation and this requires concentration. An analyst carrying out an interview should look at the work area for such items as tidiness, full or empty filing trays, layout, lighting levels, noise, temperature, supervision and number of telephones. This type of information can be very valuable as a means of verifying comments made by the staff regarding the work area.

5.9.3 Record searching

Quantitative information such as volume, frequencies, seasonality, trends and ratios is essential data for analysis and design. Probably the only reliable method of obtaining this type of information is to conduct a record search. This will also help to confirm any estimates provided by managers or staff.

The problem of missing the obvious was mentioned earlier. One manifestation of this problem is in missing 'exception' routines. By their nature, these routines do not occur very often, but they are important. When interviewing the user, the analyst will not only investigate and record the normal procedures, but also ask questions about exception procedures. Analysing historic formal documents (forms, invoices, production control sheets, etc.) is often a very fruitful source of indication of what exceptions occur.

5.9.4 Special-purpose records

It may not be possible to find out all quantitative information to the accuracy required by record searching. The type of information missing might be, for example, the number of telephone conversations in a given period or the number of times that files have to be consulted. This type of information can be collected on a special form which is completed by staff for a limited period. Other ways of collecting information are by taking a photocopy or carbon copy of a document or by making a mark in a special column added to an existing form. The completion of a form takes time, so the analyst must be sure that the information collected is essential and then convince the people who provide it that it is worth their time and effort. The information should be collected for as short a time as is necessary to obtain the required information.

5.9.5 Sampling

Two types of sampling are introduced here: statistical sampling and activity sampling.

Statistical sampling can involve record searching, which has already been described as a method of obtaining quantitative information such as volume, trends and ratios. It could also include timing parts of the work, e.g. the time taken to process a document. Special-purpose records may have to be used. The resulting statistics can show trends (e.g. workload increasing or decreasing in a specific area) or ratios (e.g. proportion of sales invoices to credit notes per week). This method can confirm management or user estimates and verify information given at interviews.

The analyst should be trained in statistical analysis before drawing conclusions from samples, or advice should be taken from a statistician. Care should be taken that samples are truly representative; obviously, results from a non-representative sample could prove disastrous. The analyst should also take care when estimating volumes and field sizes from samples and remember to cater for exceptional circumstances.

Activity sampling involves observing the state of an activity at predetermined times. However, the times will be random so that those involved in the operation do not know in advance when they will be observed. The method can be used where it is not practical to measure the total number of documents or activities that occur because of their high volume and the time involved. By using activity sampling methods, it is possible to obtain an estimate of documents or activities to a known accuracy.

5.9.6 Questionnaires

There are two major problems associated with question-naires:

- ❑ designing them so that they may be completed easily and accurately;
- ❑ getting them filled in, in sufficient numbers to make the effort worthwhile.

Filling in forms is not a popular activity with many people, and this means that unless there is some good reason to complete the questionnaire, it will end up at the bottom of the filing tray. A questionnaire should be used if similar information is required from a large number of respondents and/or from a number of geographically remote locations. If a questionnaire is deemed the best method, the analyst must be quite clear about the purpose, the respondents, their level of understanding, intelligence and interest, and the timing. Designing the questionnaire requires considerable skill to ensure that the respondents can read through the questionnaire easily and quickly without becoming suspicious or frustrated. The technique should be used sparingly as it is restrictive in the way answers may be given and presents the following problems:

- ❑ difficult to phrase unambiguously;
- ❑ results depend entirely on the interpretation of the recipient;
- ❑ answers may be inaccurate, careless or indeed deliberately misleading;
- ❑ follow-up questions are precluded.

A limited field test is recommended before sending out the questionnaire (the analyst could try out the questionnaire on a few colleagues first, and then on a small group of users before sending it out, after refinement, to the whole of the target group.).

Occasionally, a questionnaire may be designed to be completed by the people to be interviewed, before the interview takes place. This may save the time of the analyst, or the analysis team, and may also provide sets of comparable facts from various interviewees.

5.9.7 Interviewing

Interviewing is by far the most common method used in fact-finding. It brings the analyst into contact with users at all levels and provides an opportunity to listen to their opinions about the existing system, its associated problems and possible solutions. At the same time as carrying out interviews, the analyst

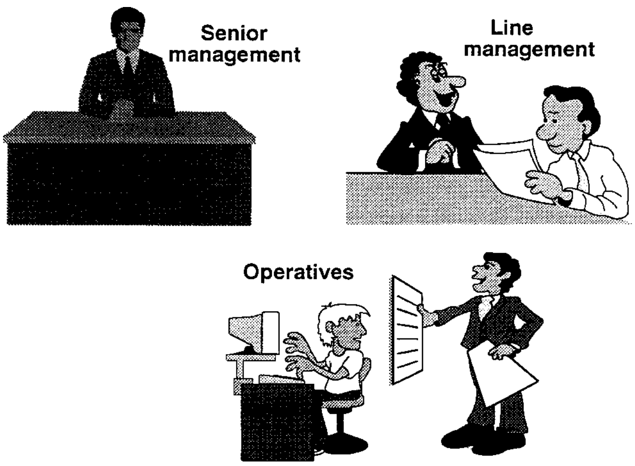


Figure 5.7 Different levels of user.

can exercise his or her powers of observation. Interviews take a good deal of time for everyone involved, so it is essential that they are well planned. Conducting an interview requires a considerable amount of skill. If conducted poorly, it can create problems, meaning that a lot of effort is subsequently required to repair damaged relationships.

There are three levels of user, as illustrated in Figure 5.7. The analyst must remember that people are very territorial and respect this aspect when planning interviews. The analyst should start at the top of the management hierarchy in the area under consideration and obtain permission to speak to the other staff. Certain types of question can only be answered correctly at certain levels, for instance questions about future trends can only be answered correctly by senior management. They can tell the analyst what improvements they seek in the organisation's control and planning information, along with the associated timing requirements.

From middle management the analyst can obtain more detailed requirements on the control and planning information, the reports required and their accuracy in association with their timing requirements. From operations staff the analyst can obtain precise details of how day-to-day procedures work, which may not be the same as the way their managers think they work.

5.9.8 Workshops

Workshops are meetings of a particular format, held to achieve quickly a particular purpose within a systems development project. They involve both end-user (customer) and IT staff and may be held to: obtain decisions; extract information; obtain agreement; establish commitment; secure approval. Often facilitated workshops are referred to as Joint Requirements Planning (JRP) or Joint Application Development (JAD) workshops within IT developments. The former is concerned with requirements definition. The use of an independent facilitator (moderator) to manage the process of the workshop is an important aspect of making workshops successful. Such a person can ensure that the workshop stays on track and runs to time, and that the results of the workshop are properly recorded. The facilitator would be involved in: planning the workshop; running the workshop session; ensuring that results were documented and circulated after the workshop. It is the facilitator who ensures that all participants are given the opportunity to contribute to the work of the facilitated workshop. The facilitator does not contribute to the content of the workshop, the role being very much one of ‘oiling the wheels’ of

5.10 User analysis

5.10.1 Introduction

If user aspects are to be taken into account in system design, particular analysis work is essential.

Work practice modelling constitutes a mapping of the BAM, discussed earlier in this chapter, on to the proposed organisational structure defined within each proposed solution (business system option). In considering the management structure, user roles and geography of the organisation, the WPM defines:

❏ who will carry out each activity;
❏ where each activity will be carried out.

5.10.2 User investigation

The system development team will inevitably require a detailed understanding of the users and their capabilities within the affected area of the organisation. This has obvious implications in the area of end-user job design and an insight into the skills of the various users will form a crucial part of dialogue design, since different types of user have different requirements in the way in which the automated system is presented to them.

Initial observation and fact-finding in this area result in the creation of the *user catalogue*. An example of a user catalogue is shown in Figure 5.8.

More detailed information should also be documented on the following:

❏ tasks undertaken by users;
❏ goals of individuals and resolution of conflicts between goals;

User Catalogue	MCC	
Job Title	**Tasks**	**Comments**
Branch manager	Staffing Budgetary control Take service booking Customer collects car Management reports...	
Reception controller	Take service booking Customer collects car Daily service reports Staff holiday control Prepare invoice....	
Service receptionist	Take service booking Customer collects car Prepare invoice....	
Mechanic	Carry out service Record time and materials Handover car....	

Figure 5.8 A user catalogue.

- ❑ constraints on tasks;
- ❑ performance feedback;
- ❑ sources of help;
- ❑ interruptions and bottlenecks;
- ❑ variations in workload;
- ❑ differences in work organisation across user locations;
- ❑ flexibility and autonomy.

During this phase of analysis, it is also important for the development team to classify the users according to:

- ❑ level of IT capability: dependent on other system usage;
- ❑ expectations of the new system: high (helpful), low (hindrance);
- ❑ degree of dedication: regular, casual, uncontrolled (i.e. ATMs);
- ❑ organisation culture and standards: job specification standards may be affected.

Other issues that may be relevant in coming to a detailed understanding of the users and their environment are:

- ❑ physical capabilities;
- ❑ language issues;
- ❑ knowledge needed to complete a task successfully;
- ❑ training received (and, ultimately, training needed in the light of the above);

- ❑ organisational position;
- ❑ other system usage.

The acquisition of detailed information about users and the way in which they are currently organised will impact heavily on any recommendations the development team propose within any solutions. Radical reorganisation to suit the proposed automated system more appropriately will have fundamental results in the impact analysis of each proposed solution and may also have considerable impact on the associated cost–benefit analysis.

However, where such reorganisation is the optimum solution, the results of user analysis need to be carried forward into the production of a WPM, which will map the various business activities on to the user roles identified from the above work.

5.10.3 Work practice model

The following activities form a guide to the development of the work practice model (WPM):

- ❑ define organisation structure and user roles;
- ❑ specify basic tasks;
- ❑ specify interactions between tasks;
- ❑ allocate tasks to user roles;
- ❑ specify interactions between user roles and the IT system;
- ❑ allocate user roles to user job specifications.

Individual jobs can be designed on completion of the above activities.

5.11 Procedures, data and documentation

5.11.1 Procedures

It is possible, and often useful, to make distinctions between what people actually do to carry out a job and what they do to prove that they have done it. These can be described as normal procedures and control procedures.

Normal procedures are concerned with handling inputs, referring to files as needed and producing outputs, following the conventional definition of data processing.

Control procedures are concerned with security, accuracy and quality. For instance, any form of batch control, involving group totals or document counts, is a control procedure. If any form of stock checking is practiced, which requires reconciliation of physical stock with recorded stock, this is also considered to be a control procedure.

There are also abnormal or exceptions procedures. These can arise because of errors or omissions in the input, because of some special report required, or even because of some regular but seasonal peak loads. There may be an exception procedure, which applies when a customer has exceeded a credit limit, and other procedures which are only applied at the year end.

Often, when exception procedures are discovered, the analyst will be told about the problems caused and how they are solved, but may not be informed as to the real problem causing the abnormal effects. This is something the analysts have to identify for themselves.

The area of exception procedures can be quite important to a new system, because the 80/20 rule can apply. This states that 80% of the volume of work done may be handled by 20% of the procedures and, conversely, that the other 20% of the work requires 80% of the procedures to handle it. In terms of computer programming, this could mean that 80% of the programs may only be there to handle 20% of the workload.

5.11.2 Data

Gathering information about the data handled by an existing system falls neatly into two categories:

- ❑ *What*? represents the actual documents handled.
- ❑ *How much*? refers to volume, frequencies, etc.

Documents are of three types:

- ❑ input;
- ❑ files;
- ❑ output.

Whenever possible, the analyst should obtain an example of each document used. Even a photocopy is better than nothing. Then, the analyst and the person who uses the documentation can establish its exact purpose between them.

Many documents include preprinted headings or boxes. The analyst must ask what each section is used for, who fills it in and if it is all completed when it arrives. If it has some kind of date on it, e.g. July sales analysis by representative, the analyst should ask when it is actually made available to the recipient. He or she should check, for each heading or box, the actual use made of it, which may be quite different from the use implied.

If some of the data received as input or produced as output is verbal, such as telephoned orders, then this must be identified and notes made of the content of the verbal message.

The data and procedures so far considered may be classed as *operational*, i.e. required for the day-to-day running of the business. There is, however, a second set of requirements to meet the 'what if' questions of management, e.g. 'What would be the effect of a price increase?' or 'What would be the effect of a fire in the warehouse?'

The analyst must determine what type of questions management might ask, how significant they are to the organisation and whether the data is available to answer such questions. It is not, at this stage, the analyst's job to provide a system which will do this. That is part of the design process.

5.11.3 Documentation

It is important, for subsequent reference, to make detailed notes of data handled, including such information as the actual layout of the data, whether it is alphabetical or numerical, whether it conforms to any definable pattern, from whom it comes, etc. Copies of completed documents currently in use should be kept as part of the system documentation.

In addition to investigating and recording the content of documents used in the system, the analyst needs to know how many of each type of document, and thus each type of transaction, are used.

A standard document for recording the points covered during a meeting or interview is advisable. Figure 5.9 shows possible contents for a discussion record. As with any good analysis form, it will act as a check to ensure that all the information which should be recorded, is recorded. The cross-reference column may be used to enter references to other documents collected or referred to at the discussion (e.g. copies of user forms).

The analyst can also use documentation methods which have been developed by Organisation and Methods (O&M) officers, when these are needed. The simple cross-reference chart, which shows how two different sets of factors interact, is a common example. Documentation standards

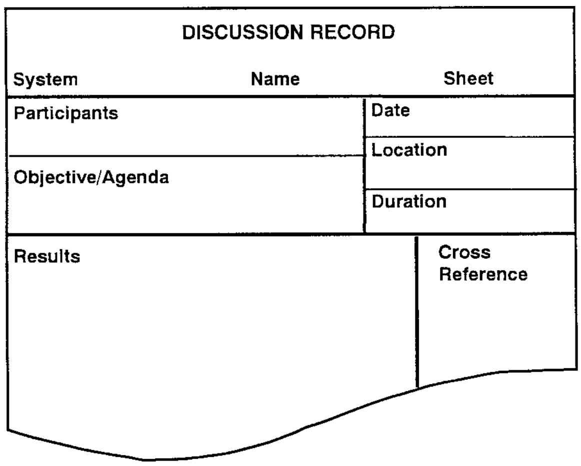


Figure 5.9 Example of a discussion record.

User / Document	Requester	Buyer	Order clerk	Purch. ledg. clerk	Goods inwards	Accountant
Purchase request	1	2	3			
Order			1	2	3	
Goods inward note			2	3	1	
Supplier invoice	3	1	2	4		
Statement		1		2		3
Remittance advice				1		2
Cheque				1		2

Figure 5.10 Example of a grid chart.

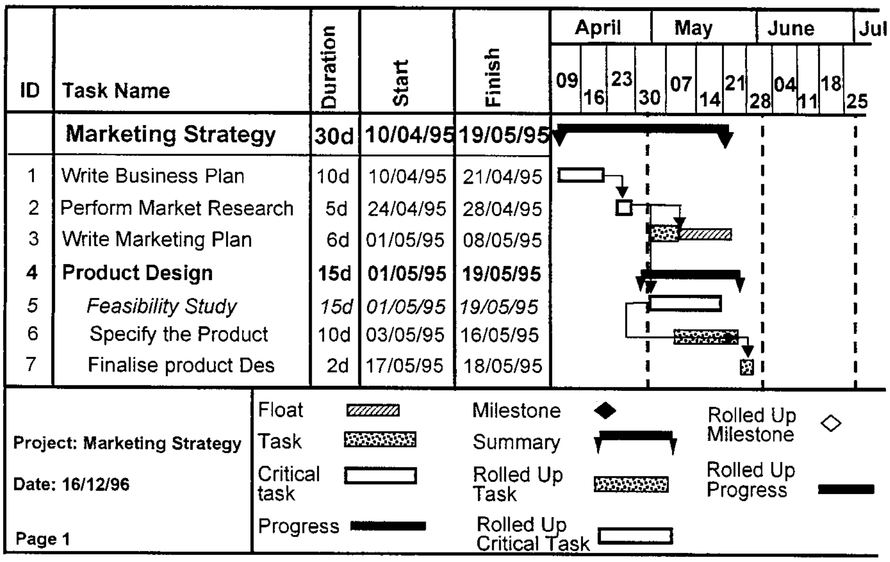


Figure 5.11 Example of a Gantt chart.

commonly refer to this as a grid chart. Figure 5.10 illustrates a grid chart in which the paths of several documents are traced around the people who raise and/or handle them.

Gantt charts are often useful in relating activities to time, particularly if they are used to relate plans to actual achievements. The example in Figure 5.11 shows details of a Gantt chart for the first stages of marketing a new product.

If an analyst is in doubt about remembering questions to which answers are needed, a general-purpose questionnaire can be created (see Section 5.9.6).

5.12 Summary

This chapter has concentrated on the early stage of analysis, i.e. the *investigation*. Figure 5.1 illustrates a feature of information systems development that should never be forgotten: time spent at the start of the project, ensuring that the objectives of the new system are correctly identified and that the user's needs are met, will be reflected in lower overall development and maintenance costs.

These are numerous problems in the general area of analysis, some of which have been identified. The following chapters show ways in which analysts can avoid or counter some of these problems. Others can be minimised if the analyst is aware that they may exist.

This chapter includes some details of the background knowledge needed before starting an investigation and has spelt out the facts about the system which must be identified and recorded. Specific methods of recording the information collected, and ways of finding out that information, have been covered. In particular, techniques of Business Activity Modelling and Work Practice Modelling have been introduced.

Finally, in order to obtain the maximum amount of information from the users, the analyst should ask questions that do not lead to monosyllabic answers. This can be done by using open questions beginning with *what, why, where, when, how, who* as recorded by Rudyard Kipling in the short stanza:

I keep six honest serving-men
(They taught me all I knew)
Their names are What and Why and When
And How and Where and Who.

Exercises

5.1 Name four methods of investigation (fact finding) that the systems analyst could use. Give at least one advantage and one disadvantage of each.

5.2 How would you, as an analyst, justify spending a considerable amount of time eliciting and checking user requirements before beginning to build code for a system?

5.3 What is a requirements catalogue? Give four things you would record about every requirement.

5.4 What are functional requirements. What are non-functional requirements?

5.5 What is Business Activity Modelling? What is Work Practice Modelling? What are the advantages of these approaches?

6
Data flow diagrams

Introduction

Pictorial representations of business activities are generally more effective than narrative ones. Flowcharts have been used for many years as an illustration of a system or part of a system; however, unless there is a standard to which flowcharts are drawn, they become the 'art of the individual'. Flowcharts portray the *procedures* that are taking place, whereas Data Flow Diagrams (DFDs) approach the system from the *data* aspect. Systems design has borrowed from the traditional approaches of engineering to develop techniques known as *structured design*. This structured approach allows simple definitions at a high level and *levelled* definitions at lower levels as necessary. The idea that the human brain cannot grasp a complex whole in fine details, all at one time, is behind the levelled approach. The brain can grasp an overall concept, without detail, or can perceive all the fine detail of a limited aspect of the concept. Systems design also recognises that emphasis can usefully be placed on the *data* that is being processed, rather than on the processes themselves. Stressing the processes rather than the data is an approach which asks '*Here is a tool, how can we use it?*' rather than the more flexible approach which asks '*Here is the data, what needs to be done to it?*'.

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

There are still conventions to be followed when producing DFDs, resulting in a high level picture which is then expanded at lower levels.

6.1 Definition of a data flow diagram

6.1.1 Data flow diagram symbols

DFD symbols are shown in Figure 6.1.

An *external entity* is a source or recipient of data that is outside the boundary of investigation.

A *process* rectangle indicates that some activity is taking place on the data. Data flows into a process, is transformed, then flows out of the process. The wording within the rectangle should be a simple imperative sentence. Any decisions are within the process, there are no 'decision' symbols in a DFD. The boxes are numbered for reference to lower level DFDs. The name at the top of the box is that of the department or person responsible for the process.

A *data store* (the open-ended rectangle) indicates a repository of data that is accessed or changed. Normally these will be files, computer or physical, but can also be disks, telephone calls, the Internet, a colleague, etc., i.e. anywhere information is extracted from or placed into by the process. Each unique data store is given a number which can cross-reference a list of data items or fields contained in that store. The letter 'M' indicates a manual data store, 'D' a computer data store and 'T' a transient or temporary data store.

The *data flow* symbol is a line with an arrow showing the direction of flow. It should be named using words that are understood within the organisation or department describing the data.

6.1.2 An example

The example DFD shown in Figure 6.2 is that of a training company. Customers who require training courses to be run on their own premises require a quotation for the course. Details of the appropriate course, fees, etc., are extracted from the course file. Details of the customer and the request are stored on the customer file.

6.1.3 External entity symbol

The fundamental purpose of this symbol, which is either a *source* of data (the enquiry) or a *sink* (the quotation) is to indicate that, whatever happens at the end of the data flow, it is not the concern of the activities being studied or designed in the DFD. It can be seen from the above example that the way in which a DFD is bounded is shown by the data flowing into the area from the source(s) and the data flowing out to the sink(s). The data flow to or from an external entity crosses the boundary of the area being described. In lower level DFDs it is not necessary to show the sources or sinks of data flows. External entities may be people, departments, functions or other groups external to the area being described by the DFD.

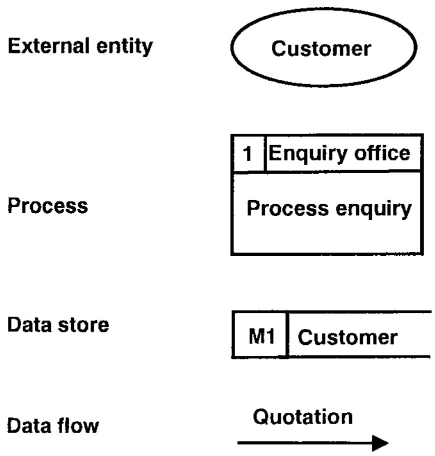


Figure 6.1 DFD symbols.

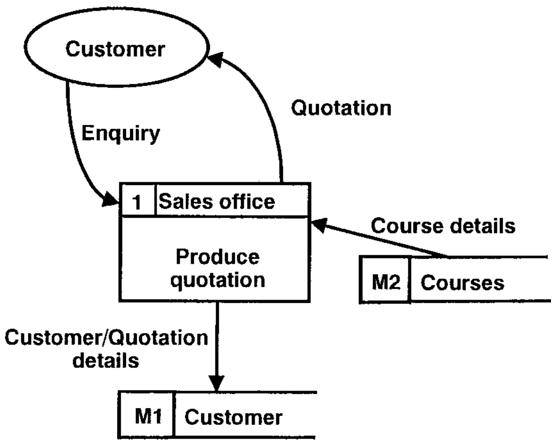


Figure 6.2 Example of DFD.

6.1.4 Process symbol

In a DFD the rectangle implies that some activity takes place, transforming the input data in some way to produce the output data. Details of the enquiry (customer, course required, dates, location, number of attendees) are input. These details, together with prices, become the output, i.e. the quotation.

There must be no increase or decrease in the data as a result of the process. If the total of the inputs does not equal the total of the outputs (including any data stores referred to or added to) in all processes, the DFD is incomplete. In

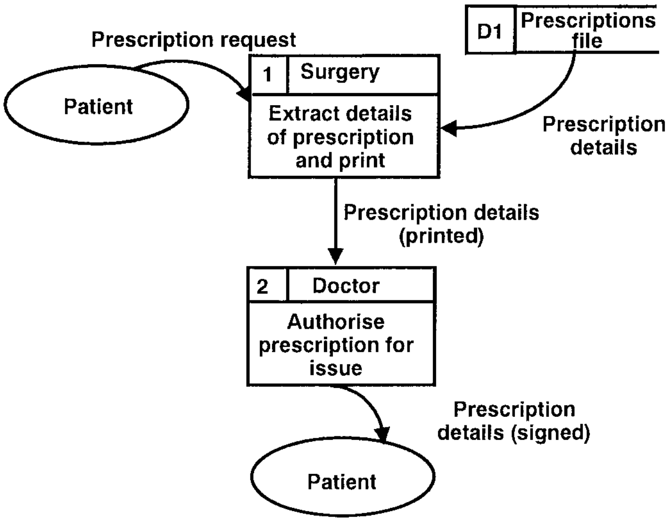


Figure 6.3 Prescription data flow diagram.

INPUT	**Prescription request**	**Extract from Prescriptions File**
	Name	**Drugs**
	Address	**Quantity to be issued**
	Telephone No.	
	Ailment	
OUTPUT	**Prescription details (printed)**	
	Name	
	Address	
	Ailment	
	Drugs	
	Quantity to be issued	

Figure 6.4 The data items related to the data flows.

the above example, customer, course required, dates, location and number of attendees are input. These details are also output (on the quotation) plus the fees which are extracted from the data store courses.

For example, in Figures 6.3 and 6.4 the total of the inputs does not equal the total of the outputs. Telephone number was input but was not output or stored in any file. If not used, the telephone number should not be input. If it is required the analyst needs to find out what happens to it, as the DFD is not complete.

Any decisions taken are contained within a process. There is no separate decision symbol.

In the prescription example, the signature of a doctor must be added to the prescription before it is handed out. This does not affect the net flow of the data because it is not a new data item but simply a *control function*. Controls are neither usually needed nor appropriate on a DFD. The purpose of a DFD is to show what happens, not when, under what external influences, or why it happens.

6.1.5 Data store symbol

There are only two justifications for data stores in any system. One is that data exists which will be required frequently for output, but which would be costly and time consuming to input each time. Therefore, *reference* files are created. The second reason is that data exists which is not required currently by the process but which will be required, either in another process or at a later date. *Time-shifting* storage files accommodate this requirement. All data stores fall into one or both of these categories. This will have some effect on data analysis, which is considered in Chapters 9 (Entity Analysis) and 10 (Relational Data Analysis).

When a data store is indicated on a DFD, the flowlines to or from the data store can be of two types. When the data store is used only for reference, the flowline is in one direction from the data store to the process (Figure 6.5).

When the data store is updated, two data flowlines are possible but it is normal to show only the net flow either to or from the data store.

If the data store is updated *in situ*, then the record to be updated (either a paper record or a computer record) is extracted first. However, the net data flow is *to* the data store, containing only the update information, so that is shown and the extract is ignored as being an unimportant detail of the processing mechanism (Figure 6.6).

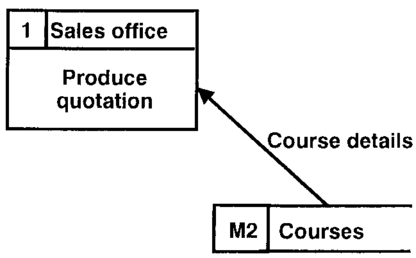


Figure 6.5 Data store symbol.

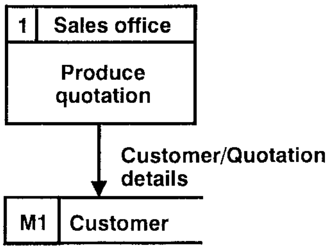


Figure 6.6 Arrow indicating the net data flow.

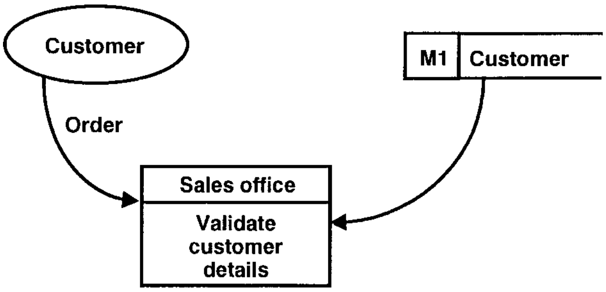


Figure 6.7 Unnamed data flow.

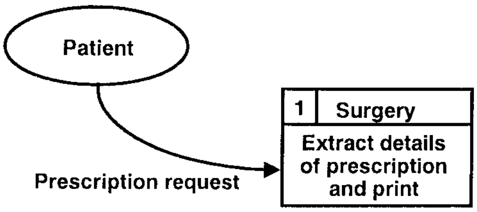


Figure 6.8 Data flowline.

It is not always necessary to name flows to and from data stores. For example, when validating customer information against a customer data store, it is only necessary to show a data flowline from the data store to the process (Figure 6.7).

6.1.6 Data pipeline or data flowline

No alteration to data can take place within a flowline. The data that leaves one process is exactly that which arrives at the next process. An arrow, usually at the end of the flowline, indicates direction of flow (Figure 6.8).

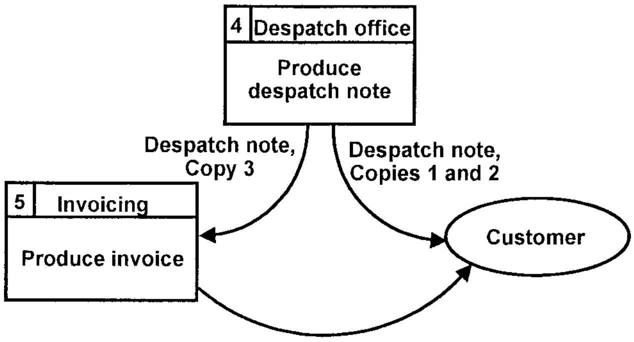


Figure 6.9 Example of branching data flows.

Sometimes it is convenient to have *branching* data flows (Figure 6.9). This may happen when a particular process carries out some work and produces a data output which is copied to two or more processes. For instance, in clerical terms, a two-part set could be split after completion, and this could be shown by a single data flow leaving one symbol and splitting to arrive at two other symbols. In such a case, both lines formed after the split must be identified (e.g. original and copy).

6.1.7 Summary of conventions

Rectangles are processes which show that some data manipulation or processing takes place. The net flows to and from a single process must balance.

Lines are data flows, often net data flows if they show data moving between a process and a data store. It may be a useful concept to visualise data flows (lines on the DFD) as data structures in motion, and data stores as data structures at rest.

Data store symbols represent data stores and data structures at rest. They may be connected to a single process or to more than one, as appropriate.

Terminators are used to show sources and sinks, which may be people, departments, functions or other groups external to the area being described by the DFD.

All processes are numbered and given meaningful names. In a high-level physical DFD the name may simply be that of a department. All flowlines (data flows) are given meaningful names (except where the flow is to or from a data store, when they may be omitted).

A description of the content of the input data flows must contain everything required to produce a description of the content of the output flows. Nothing else can be present in the output.

6.2 Levels

The DFD in Figure 6.3 shows the issue of a repeat prescription. It is a simple illustration, but the model it presents is quite genuine and the processes shown all occur in the way that they are shown. Each process is a summary of the actual activities performed, and each one can be expanded into another DFD (or miniature specification) to show in greater detail how the data is processed. Figure 6.10 shows an expansion of box 1, labelled Surgery, in Figure 6.3.

At this level there are four processes. These may or may not be carried out by different people, but each represents a distinct activity. There is also an acknowledgement that some errors will occur and will need correction. Leaving symbol 1.1 there is an 'error' data flow. This makes the net flow through symbol 1.1 correct. One may expect to find details of how errors can occur, can be detected and can be dealt with, in a further expansion of that symbol. The sources and sinks of the data flows entering or leaving the DFD can be found on the next higher level of the DFD, and they should match.

The prescriptions data store symbol is repeated here and is also shown on the original higher level DFD. This is simply to make clear where the prescription details come from.

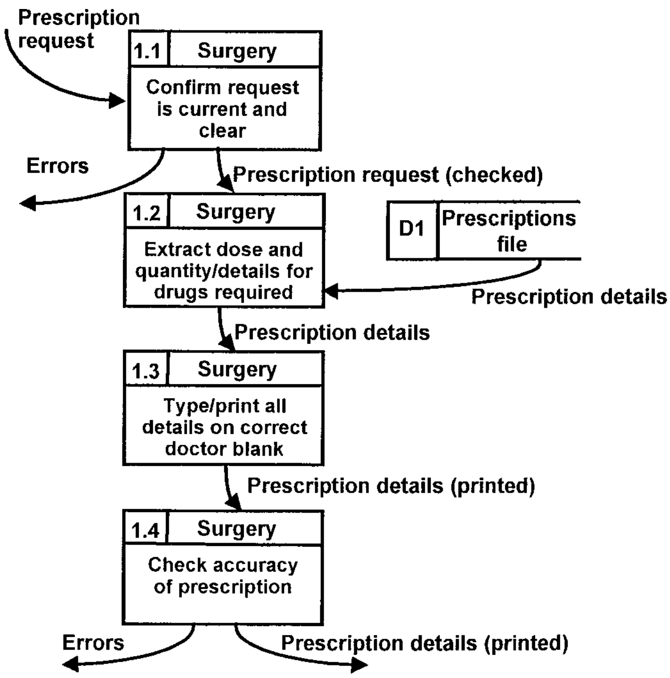


Figure 6.10 Lower level data flow diagram.

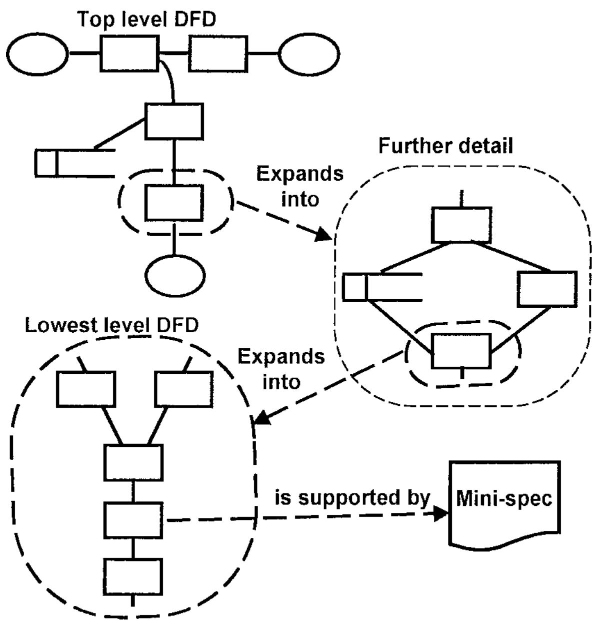


Figure 6.11 Levelled breakdown of data flows.

Even this very simple example illustrates the idea that a DFD can be produced at a high level, and then any of the processes can be expanded into a more detailed description with a lower level DFD, as shown in Figure 6.11.

The important point is that a lower level DFD must have the same data flows entering and leaving it as has the process symbol at the higher level. This can be continued to further levels if necessary, and each lowest level process (on whatever level of DFD it appears) is supported by a miniature specification ('mini-spec') of the process.

In a *mini-spec*, structured English or *pseudocode* is used, possibly together with decision tables and flowcharts where these help to structure the logic. These and other forms of procedure description are explained in Chapter 8 (Procedure Specification). They are the lowest level description of a process symbol. Note that no new information is supplied by the lower levels: each is simply a more detailed statement of what the summary shows at the higher level.

6.2.1 How many levels?

There are no rules about the number of levels of DFD used. This is wholly dictated by the application and the analyst. In some cases, a single DFD will show all that is needed, supported by one mini-spec per process. In other cases, two, three or more levels of DFD may be necessary. The important point is that the mini-spec should be mini, i.e. small, ideally about half a page.

There will always be a temptation to include too much on a single level of DFD. If the analyst finds that there are *more than ten* processes on a single DFD, it may be worthwhile reviewing it to see whether the basic functions are being clouded by excessive detail. Ten is an arbitrary number, of course, and five may be too many in one case, while 15 may be too few in another. The warning remains: do not cloud the basic processes with too much detail. The whole point of structuring is to allow the analyst, the user and designer to comprehend the diagrams produced, to detect errors or omissions and thus to reach a common understanding of the system.

6.3 Developing a data flow diagram

6.3.1 Identifying key data flows

The first task for the analyst is to identify the 'key' or important data flows. These can be on documents, forms, reports, magnetic media, data links, e-mail, screens, etc. What makes a data flow important is that is has some form of unique identifier or key. For example, an order form has a unique order number, an invoice has an invoice number, and a product has a product code. If a document is important enough to the organisation to have a unique reference then it is likely to be a key data flow. However, data that is just recorded without any unique identifier, such as a fax form, a standard letter or a quality review document, is not uniquely identifiable. A standard letter, if stored on a word processor, is identifiable by its name, e.g. Lett01, but the individual occurrences are not. Lett01 might be sent to 100 people each week.

It is not necessary to be absolutely accurate in identifying key data flows, but if in doubt an item should be recorded: it is more important not to miss anything.

In a library system a member has a membership number, a book has a book number, issues and returns of books can be identified by the date and time, and a reservation for a book has a reservation number. These are all key data flows to be included. Other potential data flows include the list of members who owe fines, the list of books that have not been taken out over three years, and details of the purchase and payment for old books placed on sale by the library. All are possible data flows and as such should be recorded.

6.3.2 Creating an initial data flow diagram

Sources and sinks (external entities) in the system are identified, i.e. where the inputs and outputs come from or go to, such as departments, individuals or existing computer systems. The path of each data flow is recorded in turn

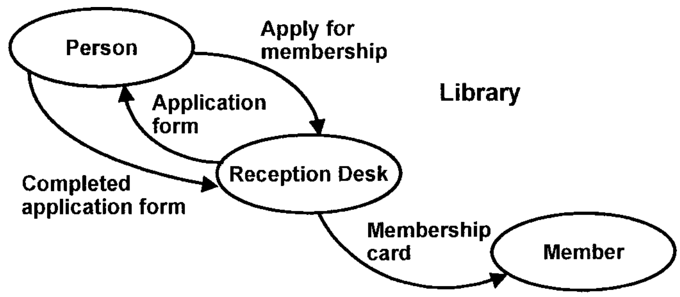


Figure 6.12 An initial data flow diagram.

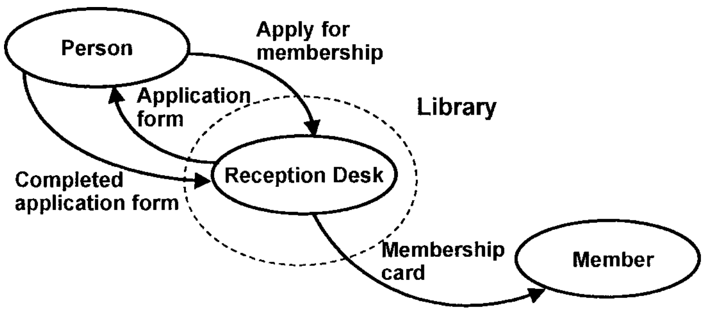


Figure 6.13 System boundary.

from source to the first functional area (department or person) that uses it, then the next functional area, until the 'life' of the data flow is complete. Functional areas and external entities are both drawn using the oval symbol. A simple document flow diagram is shown in Figure 6.12 for part of a library system where a person applies to join the library, fills in an application form, then receives a membership card.

6.3.3 Agreeing the system boundary

The diagram is reviewed with the users to determine the functional areas within the area of further investigation described by the DFD. The path of each data flow from its source to the first functional area that uses it, and through subsequent functional areas until it reaches its final sink, should be recorded. This makes it easier to identify clearly and agree with the user which functional areas are within the scope of further investigation (Figure 6.13). The 'person' who becomes a 'member' is outside the scope of investigation.

6.3.4 Identifying processes and data stores

Each functional area within the system boundary is turned into a process box (Figure 6.14). Data stores may be either added at this level or left until a lower level of DFD. If showing data stores only the major ones are included, otherwise the diagram may become too cluttered.

A person applying for membership will *trigger* the process of providing an application form. A *trigger* is the arrival of something (data) or someone (customers, members) at a functional area (reception desk, chief librarian) which causes a process to be carried out. The arrival of the completed application form at the reception desk will *trigger* the process of producing a membership card, whereas the arrival of a new stock of books at the library will *trigger* the process of cataloguing the books. Processes can also be triggered by time. For example, three times a year the chief librarian receives details of books that have not been borrowed for three years. These books are taken off the shelves and put on sale to members of the library (Figure 6.15).

6.3.5 Resource flows

Two optional symbols (see Figure 6.16) are used to show the movement of physical resources (goods rather than data). A broad arrow represents

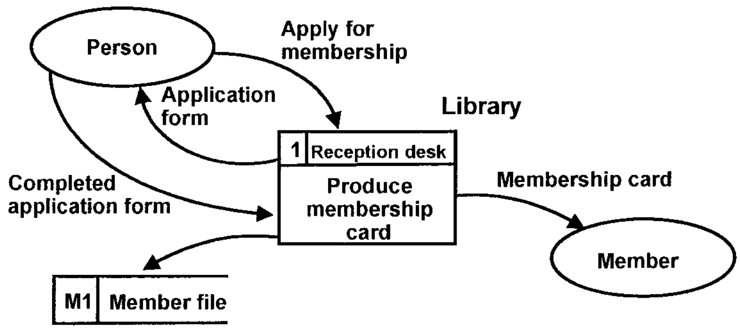


Figure 6.14 Identify processes and data stores.

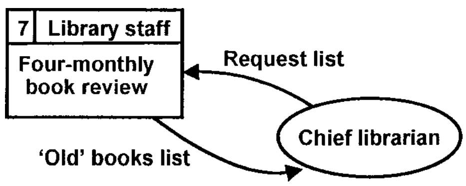


Figure 6.15 Time-triggered process.

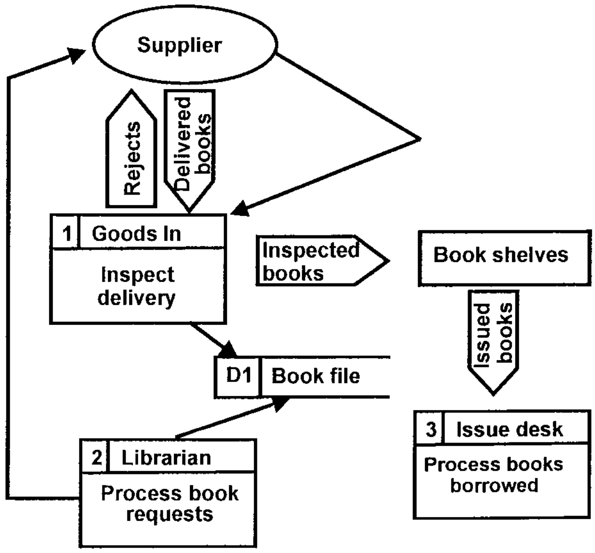


Figure 6.16 A resource flow diagram.

movement of physical resources and a closed rectangle represents a store of physical resources. In many cases an information flow will follow a resource flow, e.g. a delivery note with a delivery of goods, both of which would normally be shown on the DFD.

However, in some existing systems the actual presence of goods may be the only information available. In Figure 6.16 the process 'Inspect Delivery' is not carried out on sole receipt of the delivery note but on the receipt of the books as well. The diagram also shows what happens to the inspected books: they are placed on book shelves, rejected books being returned to the supplier. When discussing the DFD with users, resource flows on a DFD can add clarity. At the logicalisation stage of analysis, discussed in Chapter 13, the resource flows will be removed.

Figure 6.17 shows permissible connections between DFD components. For example, An external entity cannot directly access a data store, a process must exist between the external entity and the store (see Figure 6.18).

Likewise a data store cannot communicate directly with another data store without there being a process between them (Figure 6.19). There can be a connection between external entities, which is usually shown as a dotted line between the two (Figure 6.20). Figure 6.21 shows how these connections are represented.

	External entity	Process	Data store	Resource store
External entity	External data flow and resource flow only	Yes (data flow and resource flow)	No	No
Process	Yes (data flow and resource flow)	Yes (data flow and resource flow)	Yes	Resource flow only
Data store	No	Yes	No	No
Resource store	No	Resource flow only	No	No

Flows NEVER join up to each other

Figure 6.17 Permissable connections.

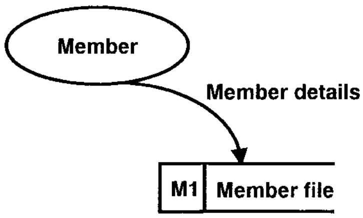


Figure 6.18 Illegal connection.

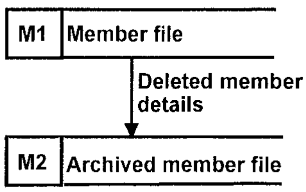


Figure 6.19 Illegal connection between data stores.

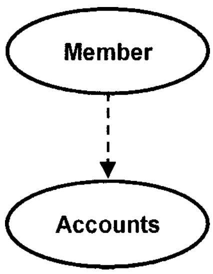


Figure 6.20 Connection between external entities.

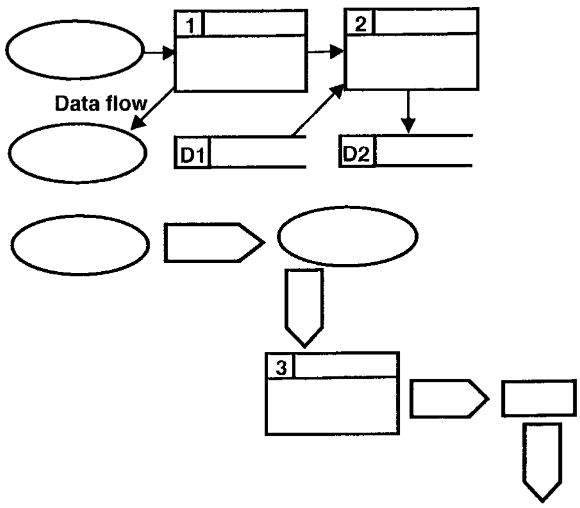


Figure 6.21 Data flow diagram connections.

6.3.6 Adding further processes

Each data store is studied to ensure that somewhere on the diagram it has a data flow going into it and a data flow coming from it. ie answering the questions how does the data get into the data store? and once there what process uses that data? (if no process uses the data, why are we storing it!!) In Figure 6.22, new member details are added to the member file, therefore there needs to be a process to remove members (as well as one to change member details such as address and telephone number).

The lines across the external entities indicate that entity appears *more than once* on the diagram. Similarly, the double bar across the data store member file indicates the same thing. This convention allows DFDs to be produced clearly without having too many lines which confusingly cross over eachother on the diagram.

In the example shown in Figure 6.23, the invoicing department sends out invoices to customers, keeping a copy themselves. When payment is received it is recorded on the copy of the invoice, which is then filed in a completed invoices file. When the invoice is raised, a copy is placed in the outstanding invoices file. When payment is received, the outstanding invoice is extracted from the file, the details of payment are recorded on the invoice and it is placed in the completed invoices file. The outstanding invoices file has the copy invoices placed into it and the copy invoice extracted from it. The completed invoices file only has a data flow into it. On further discussion with the user the analyst discovers that the completed invoice file data is

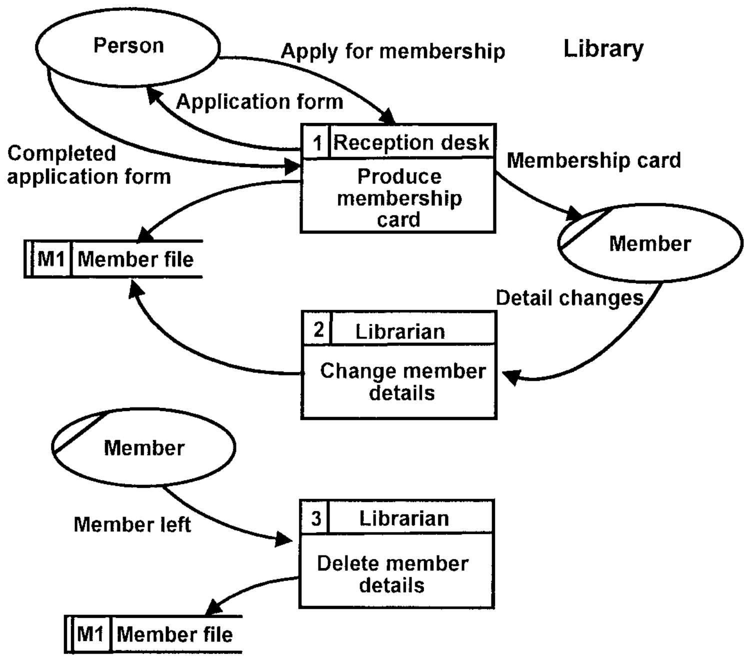


Figure 6.22 Add further progress.

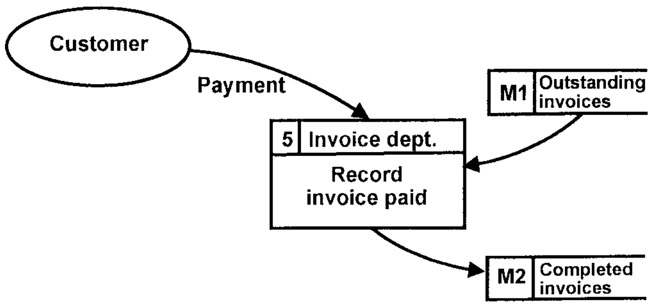


Figure 6.23 Invoicing department data flow diagram.

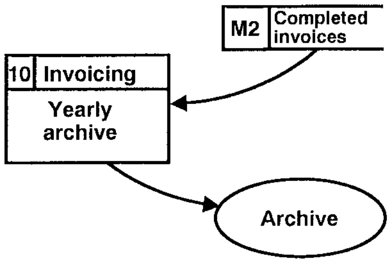


Figure 6.24 Yearly archive process.

archived once a year. This process needs to be added to the DFD. It is a time-triggered process (Figure 6.24).

6.3.7 Reviewing the first-level data flow diagram

This diagram is known as a level 1 DFD. It should be checked as follows:

- ❑ the complete life of each data flow has been recorded;
- ❑ each data store has a flow going into it and one coming out (not necessarily from the same process);
- ❑ data stores are not connected to external entities (there must be a process between);
- ❑ data flows do not disappear into a process: there should be a data flow coming out of the process;
- ❑ all names used are familiar to users.

The diagram should be checked and agreed with users before further refinement.

6.4 Lower level data flow diagrams

DFDs should be simple and readily understood. It is unlikely that one DFD will be sufficient to show all details of a system, so a set of DFDs is drawn, forming a hierarchy. The top-level diagram is called a level 1 DFD and establishes the scope of the system, the boundary being defined by the data flows to and from external entities.

The main functional areas of the system are represented by process boxes on the level 1 DFD, and each of these processes may be decomposed into a lower level DFD. The process box can be considered as a window into a diagram at a lower level in the hierarchy. The lower level diagram contains more details of the flows, data stores and processes that belong within the higher level process. Level 2 process boxes can be further decomposed to a third level if necessary, and so on (Figure 6.25).

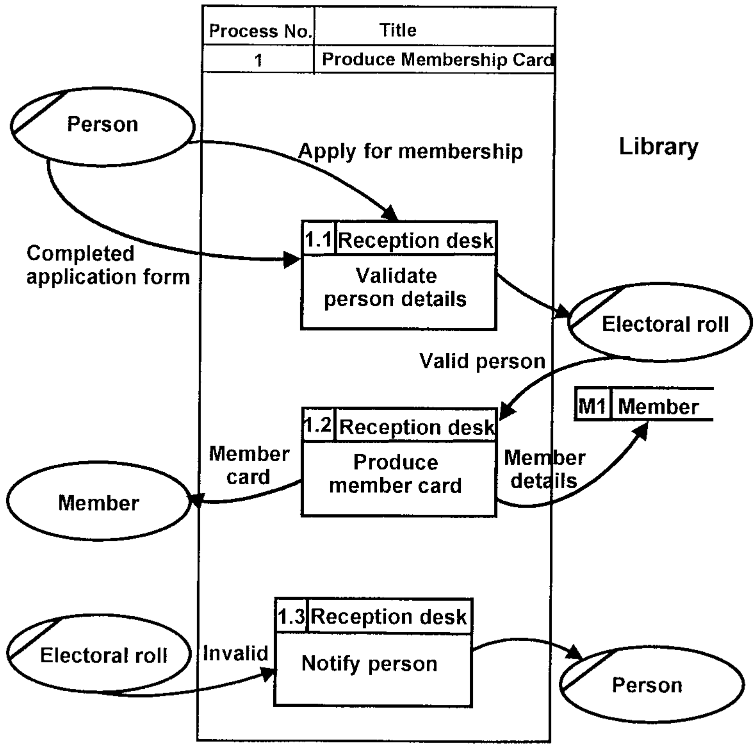


Figure 6.25 Level 2 data flow diagram.

The process box at the higher level becomes the DFD boundary at the lower level. Each process in the lower level takes the number from the higher level and adds a decimal extension, i.e. 1.1, 1.2, 1.3, etc. Data stores used elsewhere in the higher level are shown outside the boundary. Any data stores used only at the lower level would be shown inside the boundary and numbered D1/1, D1/2, D1/3, etc. External entities are always shown outside the boundary even if they only communicate with the process represented by the lower level DFD.

A process box from level 2 can again be broken down to level 3. For process boxes at level 3 numbering is extended again, e.g. if from box number 1.3 at level 2, the numbering would be 1.3.1, 1.3.2, 1.3.3, etc.

It is *recommended* that three levels, supported by mini-specs, are sufficient to define a system, but it may be found that more than three are required. As mentioned earlier, ten process boxes on a DFD should be the maximum for clarity; any more than ten may mean that a higher level DFD is required.

Any further description of a process will be defined in a mini-spec. These

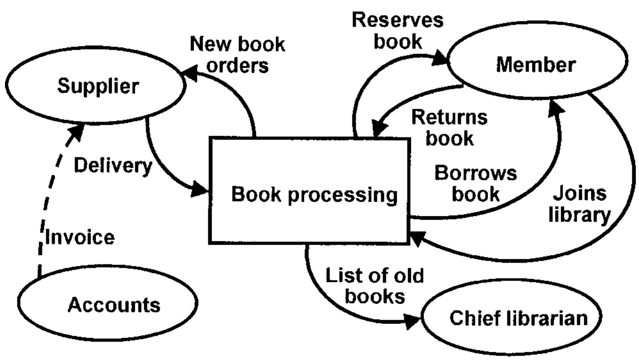


Figure 6.26 Context diagram.

can be in the form of flowcharts, decision tables, structured English or pseudocode (see Chapter 8: Procedure Specifications). They can be used for a process box at any level of DFD where only a brief (half-page) narrative is needed to define the process fully. A mini-spec is sometimes referred to as an *elementary function description* or an *elementary process description*. They should always cross-refer to the process box number that they define and contain the *same* heading.

Elementary process description

Process no.: 1.2Pr ocess name: Produce member card
Add new member details to the member file.
Produce membership card.
Produce letter A1001 and send with card to member.

6.5 Context diagrams, document flow diagrams and resource flow diagrams

If the services to be investigated are complex or the practitioner is inexperienced, the drafting of a context diagram, document flow diagram or resource flow diagram may be a useful way of getting started on the current physical DFD. A *document flow diagram* is shown in Figure 6.12 and a *resource flow diagram* in Figure 6.16. A *context diagram* is drawn showing the entire system as a single process surrounded by the external entities. The major inputs and outputs are represented as data flows. This type of diagram helps to focus on the boundary of the system to be investigated and aids agreement with the users on the scope.

The context diagram in Figure 6.26 is drawn showing the entire system as a single process surrounded by external entities. The major inputs and outputs are represented as data flows.

6.6 Summary

The production of DFDs is an iterative process. The initial DFD is the first attempt to identify key data flows, their sources and sinks. It also establishes the system boundary. From there, functional areas are expanded into processes and data stores (level 1). As more information is obtained, further definition is shown by lower level DFDs and mini-specs. The purpose of DFDs is to:

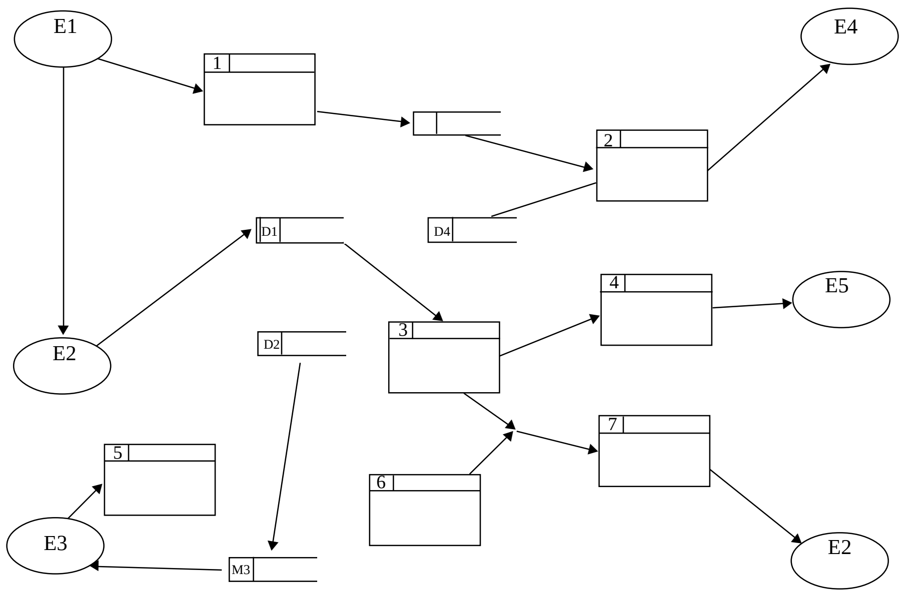
❑ use a hierarchical method to break down processes;
❑ create diagrams from the point of view of the data, rather than the process.

DFDs are flexible, can be discussed with and understood by users, and can be modified and redrawn as often as necessary. They also facilitate the process of moving from an existing system to a new required logical system, as will be seen in Chapter 13 (Logical Systems Modelling).

Exercises

6.1 DFDs use four symbols to describe processes. What are they and draw an example of each?

6.2 The following DFD has a number of errors in it. Identify the errors and state what questions you would need to ask to resolve the error. For example, store D10 has no flow into it so how does the data get stored and where does it come from?



6.3 Consider the following scenario for an Examination Administration System:

A college has an arrangement with an Examination Board for running training modules and setting and marking examinations in those modules. These modules and examinations are available publicly as well as to the college's own students. The college has a number of papers (for each module) which it has developed and will use them on more than one occasion. Since candidates may re-sit a module examination the college must keep track of the particular papers that a candidate has taken previously. Examinations are scheduled throughout the year and require the assignment of an examiner with the right skills to mark the scripts and an invigilator to administer the examination session. Lists of appropriate examiners and invigilators are kept by the college. The Examination Board audits the college periodically and the college has to submit audit information to the Board regularly.

Exam Enquiry

A potential candidate enquires about dates of examinations in a module(s). Candidate may select date and ask to make booking. If so:

a. Check to ensure that the candidate has never taken the scheduled paper before:
   i. From the EXAM BOOKING AND RESULT TABLE (history) look up for that module to see if the candidate is listed there.
   ii. If so look up the exam date.
   iii. Use this to look up the exam paper that was used for that event in the EXAM EVENT TABLE(history), eg Data Modelling4
   iv. Using this, in the EXAM EVENT TABLE, look up the exam paper that is scheduled for the exam event onto which the candidate wishes to book
   v. If the candidate has sat the particular paper before, offer him/her alternative dates with a different paper
b. Accept booking
c. Candidate confirms booking paying the fee;
d. Update Number of candidates on Exam Event Table
e. Update Exam Booking and Result Table, with Name, date of birth, Contact Details (address, phone, e-mail)

Running the Exam Sitting

1. One week before the exam event, the Invigilator Pack is set up
2. The day before the exam event, the invigilator collects 'Invigilator's Pack' (Blank Exam Papers, Answer Books, Candidate Instructions, Pens, Candidate List)

3. On the examination day, the invigilator checks who is present against the candidate list and hands out the papers
4. The invigilator reads out to the candidates, the examination instructions
5. The exam runs & the invigilator collects back the completed answer books, the examination papers (count to make sure that ALL are collected back)
6. The invigilator seals the papers in the envelope provided, signing across the seal and takes the papers to the examiner for marking.
7. The examiner takes the marked papers back to the college for results to be sent to the board which then issues certificates to those candidates who have passed.

Quarterly – Set up Exam Event

1. Book Exam Venue & Update Schedule Table and Exam Event Table
2. Book Examiner & Update Exam Event Table
3. Book Invigilator & Update Exam Event Table

Cancellation

A candidate may cancel the examination booking and transfer to another exam session at any point up to 4 weeks before the session without penalty. Full fees are forfeited within this 4 week period.

Draw a context diagram for the Examination Administration System.

6.4 For the scenario described in 6.3, draw a Level 1 DFD for the current system described. State any assumptions you have made.

6.5 Draw a level 2 DFD to describe the process of Running an Examination Sitting. State any assumptions you have made.

7
Data dictionaries

Introduction

Chapter 6 outlined how the flow of data through an information system can be represented by using Data Flow Diagrams (DFDs). The DFD shows the relationships between processes, data flows and data stores, but does not actually define any of these in detail. For the system description to be complete, these items must be rigorously defined. This chapter considers the role of the Data Dictionary (DD) in achieving this.

In its simplest form, the DD may be a paper record kept in alphabetical order, with details of the meaning, content and use of each named item. A more convenient format may be available to the analyst through an on-line terminal, using basic text or word-processing software. An even better option is a purpose-designed on-line DD program. This can carry out a wide range of processing, indexing and cross-referencing tasks to make the dictionary even more valuable.

In this chapter, the ideas and practical uses of these different forms of DD will be developed. The relationship between the use of a DD by an individual analyst and by the total development group as a basis for database implementations will be discussed. Also, the significant benefits of keeping an up-to-date DD will be identified. All analysts need to keep some form of DD.

7.1 What is a data dictionary?

A DD consists of ‘data about data’. This definition prompts further questions such as what is the ‘data’ about which information is to be kept and what information about the ‘data’ is essential to maintain a complete and usable DD?

The DD is where the analyst (in fact, the whole project team) keeps detailed information about the developing system. Details such as the length and valid values of customer number, the list of processes within a set of DFDs and the contents of all data stores included in the DFDs are held in the DD.

The DD holds information that defines explicitly the products of system development, such as DFDs and entity relationship diagrams (see Chapter 9) and, when the physical design stage is reached, their physical equivalents. Without these definitions, diagrammatic techniques such as DFDs would be inadequate. Hence the DD must, at least, have entries for:

- ❏ data flows;
- ❏ data stores;
- ❏ processes;
- ❏ entities;
- ❏ mini-specs.

Such 'things' about which data will be held are often referred to as *data objects*. It should be realised that some objects (e.g. data stores) are made up of other objects (e.g. data items or attributes). Flexible but rigorous definition of these objects requires consideration of the basic unit of data. The inclusion of definitions of basic units in the DD permits the detailing of more complex units without the need to redefine basic units. The basic unit of data is called a *data element*.

A data element is a unit of data that cannot be broken down into a smaller but still meaningful unit. The following are all examples of data elements:

- ❏ employee's surname;
- ❏ street name;
- ❏ patient number.

Data elements are thus the first group of data objects with information in the DD. An entry in the DD must fully and unambiguously describe the element.

Data elements form the detail of any system, but it is necessary to define higher level objects so that the analysis does not become cluttered with unnecessary detail. By combining various data elements, a flexible but still rigorous definition of a more complex structure, termed a *data structure*, can be derived.

A data structure is a group of related data elements and/or other data structures. For example, the structure for a patient would be:

PATIENT (the data structure being defined)
Name (a substructure)
Hospital no. (data element)
Surname of doctor (data element).

=	'consists of'
+	'and'
[ ]	enclose alternatives
|	separate alternatives written on the same line
{ }	enclose repeating elements or structures or both
( )	enclose comments, such as the 'occurs' for repeating items

Figure 7.1 Conventions for defining data structures.

Invoice = Header + {Item} + Trailer

Figure 7.2 Invoice data structures.

This definition for a patient is not complete, but it illustrates the components of a data structure. By its incompleteness, a feature of the DD is also shown, namely that the DD is a dynamic tool wherein such definitions are built as more information is gathered through the course of analysis and subsequent system design.

It is essential to have a standard for describing data structures which is not too wordy but still facilitates precise definition. This is achieved by using a small subset of relational operators, shown in Figure 7.1.

By using these operators the analyst can represent data structures very succinctly. Consider the data structure for an invoice. It consists of a header, one or more item lines and a trailer, and could be written as in Figure 7.2.

The definition assumes that header, item and trailer have already been defined in the DD, otherwise the description of the invoice would not be very informative. This invoice is a data structure composed of other data structures. By examining the DD for the definition of these substructures the original definition of the invoice can be expanded so that it consists entirely of data elements (Figure 7.3).

The essence of any dictionary definition is clarity and any method that improves this will aid the analyst. The definition of invoice in Figure 7.3 exemplifies a poor structure layout which can be avoided quite easily. A data structure of some complexity will inevitably have many components which will be punctuated using relational operators. Reading such a structure becomes a chore but by careful rearrangement the rather awkward structure of Figure 7.3 can be replaced by that of Figure 7.7.

```
Invoice    =    Customer Name + Customer Address +
                Invoice No. + Date + {Quantity +
                Description + Price + Value +
                VAT rate} + Carriage + Total Value +
                VAT + Grand total
```

Figure 7.3 Expanded invoice data structure.

```
Invoice    =    Customer Name          +
                Customer Address       +
                Invoice No.            +
                Date                   +
                 {Quantity + Description + Price +
                  Value + VAT rate}    +
                Carriage               +
                Total Value            +
                VAT                    +
                Grand total
```

Figure 7.4 Expanded invoice data structure: improved layout.

```
Description = [Narrative | Stock Code + Item Name]
```

Figure 7.5 Alternatives in a data structure.

Note, however, that parts of the structure may be substructures. For example, Description may be a data structure, which is further defined as in Figure 7.5. Here the use of the vertical bar to separate alternative choices is demonstrated.

7.2 Making a data dictionary entry

Data elements and structures are used within the definitions of all recorded objects. But what data about these objects should the DD contain? To answer this question, it is appropriate to consider the development of part of a simple paper-based DD for the various types of entry described above. A general-purpose DD entry form will be used, although more rigorous definitions would be achieved by using forms designed specifically for each type of entry being made.

7.2.1 Common parts of a data dictionary entry

All definitions in the DD have some common requirements regardless of the type of entry. These are:

- the *name* of the object created by the analyst: this should be meaningful to the analyst and especially to the users;
- the *type* of object, i.e. element, structure, flow, etc.;
- any *aliases* of the object: many objects defined in the DD will be known by more than one name by different users of the DD and also the users of the system. Indeed, users may be unaware of this and the analyst may uncover such aliases through the course of the project. For example, 'order acknowledgement' may have the aliases 'customer order acknowledgement' or 'delivery note part 3'. There is an onus on the analyst to check that a data object is not already satisfactorily defined, for his or her purposes, in the DD. Alias definition, through careless naming of objects, is to be avoided. All necessary aliases must be defined under their alias in the DD but only a cross-reference back to the primary named entry need be made;
- the *author*: this is necessary so that when a DD entry is updated, the originator of the entry can be consulted to confirm that the original meaning of the entry is not altered;
- the *date*: the creation date for the entry.

7.2.2 Data element entries

An entry for a data element will additionally consist of:

- the *definition* of a data element: to describe what the element is and what it represents;
- the *occurrence* of the element; its relationships to other entries in the DD are given. For example, the element 'address-line' appears four times as part of the structure 'Address';
- its *picture*: to detail the format of the element (and therefore its length) and also the range of values that it may have. Initially, the format may be only temporarily defined in the knowledge that it will be modified during later stages of the project. The range of values may be continuous (e.g. range 1–1000) or discontinuous (sometimes termed discrete). For example, the element 'Accommodation' may have values: H(ouse), B(ungalow), F(lat). To list every possible value in a large discontinuous range of values would not be possible within the DD. In such cases, the element is better considered to have a continuous range with each value being a key field to identify acceptable values from a data store. For example, UK postal codes begin with a two alpha character data element (part of the structure 'post-code') which identifies the distribution area office which delivers the item to the

customer. Hence the value ST would be the key field used to access information from a data store 'distribution area', and an entry would be found which gave the detail Stoke-on-Trent;

- ❑ *notes*: additional notes may contain any other useful information about the element being defined, such as design and programming considerations, e.g. stored on disk in packed decimal.

Figure 7.6 shows a completed form for a data element called 'stock-balance'.

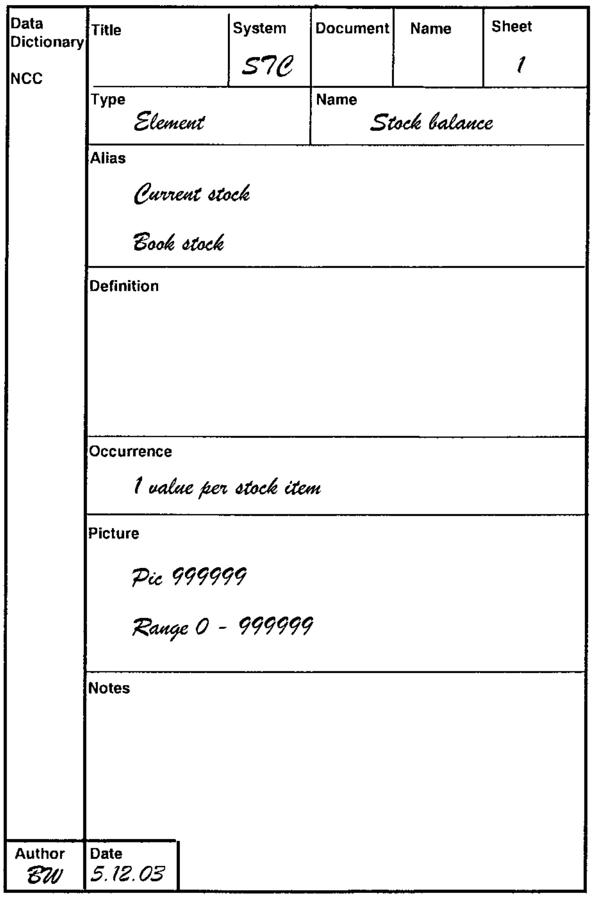
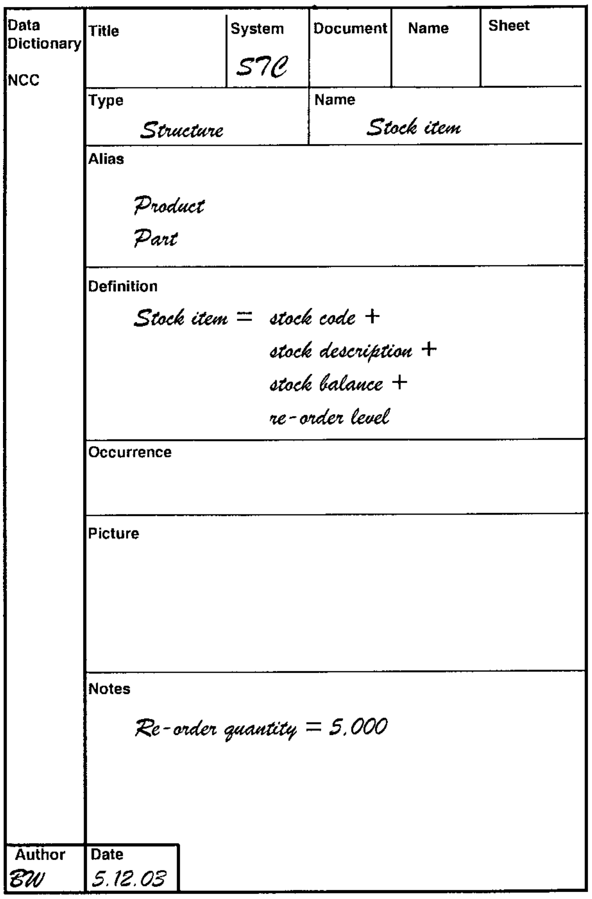

Data Dictionary
NCC
Title
System: STC
Document
Name
Sheet: 1
Type: Element
Name: Stock balance
Alias: Current stock
Book stock
Definition
Occurrence: 1 value per stock item
Picture: Pic 999999
Range 0 - 999999
Notes
Author: BW
Date: 5.12.03

Figure 7.6 Example of data element entry.

7.2.3 Data structure entries

The completion of a data structure entry requires the following:

- ❑ a *definition* in terms of the data elements and/or structures that make up the structure. The way in which this is represented was described above and is exemplified in Figure 7.7.
- ❑ *notes,* holding information such as values used as part of the control logic for a process using the structure.



Data Dictionary NCC

Title | System: STC | Document | Name | Sheet

Type: Structure | Name: Stock item

Alias:
Product
Part

Definition:
Stock item = stock code +
stock description +
stock balance +
re-order level

Occurrence

Picture

Notes:
Re-order quantity = 5,000

Author: BW | Date: 5.12.03

Figure 7.7 Example of data structure entry.

7.2.4 Data flow entries

To quote Gane and Sarson, a 'data flow is a data structure in motion'. That is to say, the flow of information consists of data elements and/or structures passing to and from processes and stores within the system or terminators outside the system boundary. The origin and destination of the flow are obviously required as part of a rigorous DD entry and could easily be accommodated in the notes section. The appropriate entries to be made for data flows (exemplified in Figure 7.8) are:

- ❑ *definition*: shows what structures, together, form the data flow;
- ❑ *occurrence*: summarises operational information, such as maximum, minimum per day/week/month, together with projected increase in flow per year. Any seasonal variations should also be included;
- ❑ *notes*: include the origin and destination of the flow together with operational information, such as response time.

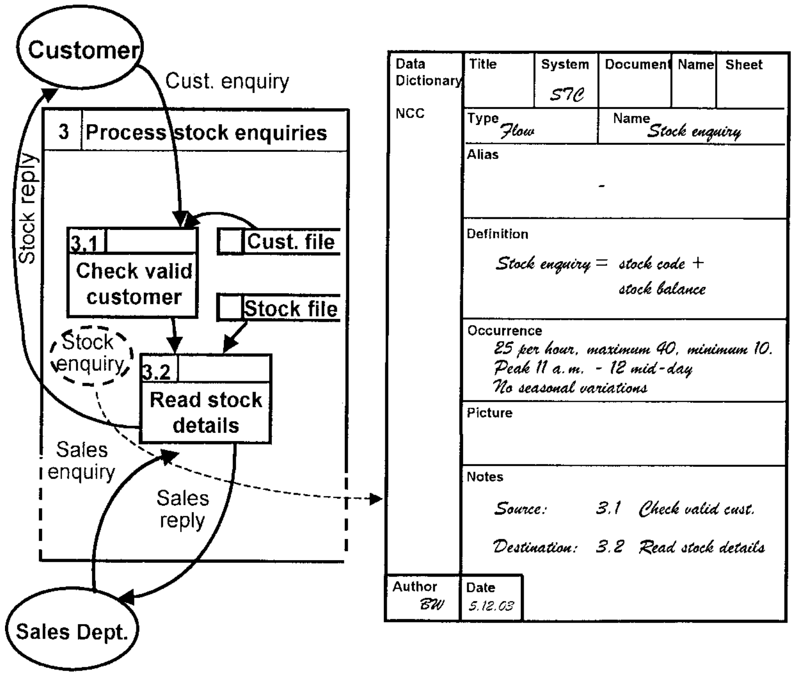


Figure 7.8 Example of data flow entry.

7.2.5 Data store entries

A 'data store is a data structure at rest', i.e. it consists of data elements and/or data structures. The DD entry needs to include the following:

- ❑ *definition*: identifies the data structures held in the data store as well as information about the organisation of the store, e.g. key fields;
- ❑ *occurrence*: details the maximum size of the store and expected growth for the store size;
- ❑ *notes*: could be used to hold details of the data flows that enter and leave the store.

A sample entry is given in Figure 7.9.

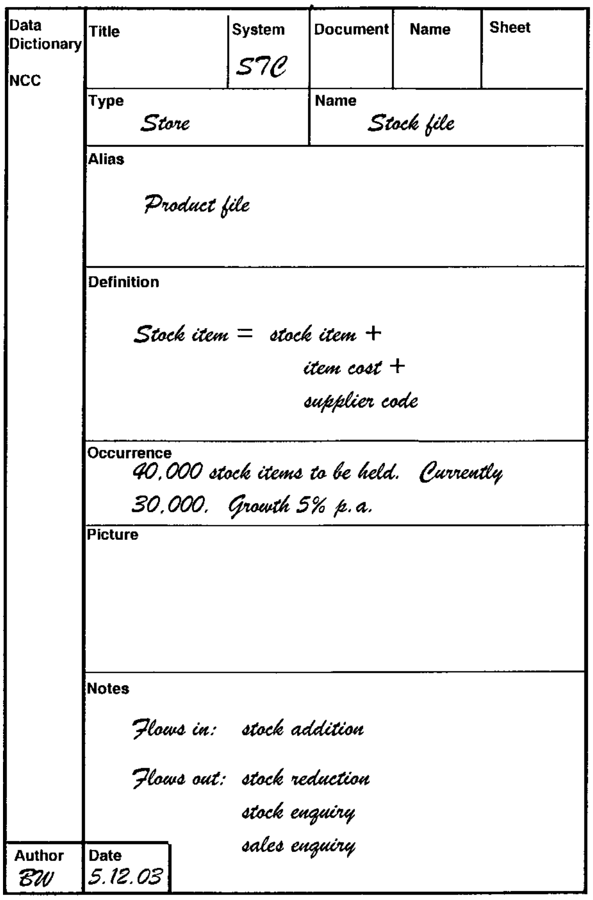


Figure 7.9 Example of data store entry.

7.2.6 Process definitions

- ❏ *Definition* of processes: succinctly describes what a process does. It may also include an unambiguous brief description of the logic, which will be expanded elsewhere in the analysis phase (see Chapter 5). A reference to the appropriate procedure specification documents must be given in the DD;
- ❏ *Notes*: hold details of all input and output flows of the process.

7.2.7 External entities (sources and sinks)

These are defined simply by a name and data flows which are linked to the entity.

7.3 Why have a data dictionary?

It should be acknowledged that this account of a paper-based DD is neither definitive nor exhaustive in its treatment. A DD is a dynamic reference document which will, as any system is developed, evolve to contain definitions of logical and physical objects together with the links between them. Organisations will rapidly evolve their own standards for entries to achieve the objectives behind the establishment of the DD. Some of these objectives will now be considered.

7.3.1 Single source

The DD forms a single definitive source of all definitions about the data upon which each project team is working and ultimately all the data in which the organisation is interested. The existence of these objects in the DD means that the analyst does not have to 'reinvent any wheels', and any confusion caused by each team having different names for the same data is minimised.

7.3.2 Reduced redundancy

If only one copy of a piece of information exists within the DD, then if the definition is modified it only requires one change to be made to the DD. Maintenance is easier and any ambiguity that might arise through the existence of conflicting definitions is eliminated. For example, suppose that 'Address' is defined in Figure 7.10, and it is discovered that the number of address lines needs to be increased from 4 to 5. As address is defined once only, this change is easily made by one amendment to the structure 'Address'. If Address were a component of many more structures, then the benefits of eliminating redundancy would be proportionately greater.

```
Address   =   {Address line} x 4
                Postcode

Customer  =   Customer Name +
              Customer Number +
              Address

Supplier  =   Supplier Name +
              Supplier Number +
              Address
```

Figure 7.10 Example of reduced redundancy.

7.3.3 Easier integration of projects

Once a DD is well established, much of the groundwork in defining data of interest to the organisation will have been carried out and the results stored in the DD. It is thus easier for new systems to be developed and integrated with existing systems.

7.3.4 Cross-referencing to other documentation

The DD holds cross-references to the system design documents such as DFDs and mini-spec as described in Chapter 8.

7.3.5 Easier transition from the logical to physical design

The DD can hold the logical data definitions, which can be used to derive the physical definitions necessary for system implementation. These physical objects would themselves constitute entries to the DD, and links between them and related entries would be identified within the entry.

7.3.6 Increased accuracy and quality

If, for example, a structure is defined when a system is being developed such that it consists of 15 elements, when this definition is rewritten as part of another document an element could be accidentally missed out. The use of a DD removes the need for such transcription as the complete structure can be referred to by its name, thus reducing the possibility of errors. The system produced will not only be of better quality but it can be produced more quickly.

7.4 The analyst's role in ensuring a usable data dictionary

The analyst has considerable responsibility to his or her colleagues to ensure that whenever an object is defined its definition is entered into the DD accurately. In addition, there is an onus to ensure that any components of the object are also present in the DD, and if already present that its definition is consistent with the use currently envisaged. Without this commitment the DD will lose one of its most critical features, that of currency. Each analyst may make their own definition of the unentered object and immediately an ambiguous situation is established.

The analyst, if not moved by this altruistic motive for immediate updating of the DD, should consider the scenario of returning to the aforementioned project several months afterwards, to enhance the system in some way, only to find that several objects are not defined and references to the appropriate DFDs are not available. A little time invested in updating the DD will be rewarded by considerable time saving in the future and hence greater efficiency. A DD which is complete is a boon to productivity, whereas incomplete it is a liability.

7.5 Automation: the path to greatest benefit

Hitherto a paper-based DD has been mentioned, but the effort required to maintain such a system is substantial. This discourages maintenance of the DD and, together with loss of DD documents, can render the DD use-less. The solution to this clerical overhead is through some form of automation. In its simplest form a word processor may be used but a far better solution is through the use of a database package or a purpose-designed DD package. Each of these will now be considered together with their relative benefits.

7.5.1 Word processor

Minimally better than a paper-based system, a word processor can only prompt for the type of entry by using the ability to include another file from disk which consists, for example, of a blank data flow entry form. A word processor will allow the insertion of new entries by 'pushing down' text that should follow the new entry, but the position in which to make the new entry has to be determined in a rather tedious fashion. A search for a particular word in the DD file can be made and, together with some examination either side of the search word, the insertion point can be found.

Whether the word-processed DD is held on a stand-alone PC, a networked system or via a terminal to a larger system, problems arise regarding currency and control over who is allowed to update the DD and when. File locking to allow updates to be made by only one person at a particular time may be possible, and will preclude others from using the DD, but it runs the risk that updates or even simple references to the DD will not be made.

As the word-processed DD file increases in size the problems of maintenance become even more laborious; therefore the use of a word processor should be restricted to the smallest of situations.

7.5.2 Database package

As the facilities required of a DD have a lot in common with general-purpose databases, it is difficult to see why database functions should not be available as part of any implementation of a DD. The database should provide support for queries and report generation, for locking data (at record level) to prevent simultaneous access and updating, as well as taking account of security considerations and any disaster recovery procedures. All users of the DD must have access to current definitions within the DD subject to essential limitations imposed by locking procedures.

Database packages allow the definition of records, data entry type checking, screen layout for data entry (equivalent to the paper-based DD entry form), and full search and updating functions. All of these functions are carried out much more efficiently by a database package than can be done with the methods so far described. Searches and updates are carried out using a key field (e.g. object name) upon which a separate index file is based. The database only needs to search a relatively short file, the index, which references the required record to be examined or updated as desired.

The limitations of a general database package and the functions that it is able to provide for a DD system depend on the tailoring of the package by the organisation as well as the inherent sophistication of the package itself. However, even the most sophisticated database package cannot fulfil all functions which a purpose-built DD package is able to do.

7.5.3 Data dictionary package

Some of the features of a DD processing package outlined below are currently available and others are features which are necessary to gain maximum benefit from this invaluable store of information, the DD.

As indicated above, any package should allow easy access for all analysts, designers and programmers to records in the DD for examination and/or updating. Enquiry of the DD would most obviously be via the name of the object, but what if the name is not known? The DD package should be able to take a description and search the dictionary for a definition of the object on the basis of keywords or strings. Access to the contents of the DD can be provided through password control and record locking in such a way that a secure single version of the DD exists at any point in time.

The DD package will produce various reports, either on screen or on paper, which permit listing of all or selected entries in full or part detail. Reports of greater complexity would also be catered for, such as the contents of a data store, requiring the DD package to expand objects to their primitive elements and derive the hidden detail in the entry for the store.

In common with the database method of implementation, a DD package would prompt for the type of entries required through preformatted screens for data entry of elements, flows, etc. However, unlike a database implementation, it should automatically check for predefinition of any objects used as a component part of a new definition. If they are not already defined, it should automatically take the analyst down the path of defining them. This completeness check will also encompass checks specific to the types of entry. Are there any flows without a source or destination? Are there any elements not used by any other object, which therefore represent redundant information? Are there any objects in a store that have no way of getting there because there are no flows feeding the store with them? Such multifaceted checks will produce a more tightly defined DD which pre-empts many problems later in the project.

As stated before, the DD is a dynamic tool which will be modified frequently. The implications of such changes impact on the many other related objects in the DD and therefore the DD package will provide a cross-referencing function to identify these objects and permit them to be updated.

When the analysis and design phases of the project lead into the programming stage, the DD is used by programmers to produce the data definitions within the programs that they are writing. Ideally the DD package takes the data definitions and can be instructed to create these data definitions in a particular target language. This prevents possible transcription errors and improves productivity, concentrating the talents of the programmer on the program logic and not the mundane aspects of the program.

It must be acknowledged that many systems have been written in the absence of an automated DD package. To encourage organisations to adopt such a package, the package should minimise the effort required to incorporate data definitions inherent in current programs. To this end, by passing the code for these programs to the DD package for scrutiny, and by telling the package the language that it is written in, the DD package can extract information about the objects used and derive DD entries for them as required.

7.6 The Data Dictionary and the CASE environment

The computer department, which for so many years has concerned itself with automating the working practices of just about every other department, is at last beginning to experience significant automation of its own. Software tools, which together make up what is known as the Computer Aided Software Engineering (CASE) environment are now widely available. These software tools take the form of a 'workbench' to assist the systems analyst or designer in the use of structured techniques, such as those presented in this book.

The facilities provided by such tools vary. Some are no more than graphics packages which aid the drawing, editing and subsequent printing of diagrams. Others provide, in addition, varying degrees of syntax and constancy checking of the diagrams, often in line with a specific structured methodology. Central to the more sophisticated tools is the DD where all project data is held. Again, the facilities available vary from tool to tool, but may include automatic extraction from diagrams and cross-referencing of the names of processes, data flows and data stores, entities (see Chapter 9) and their relationships. Supporting data may also be held; for example, a detailed description of each process, the volumes, trends and usage of the data stores, and their associated objects. In future, the DD may eventually provide common data access procedures so that each application or CASE tool does not require its own procedure. Hopefully, the DD will permit restructuring of the data such that the applications which use it require no rewriting.

Any particular software tool will usually only cover part of the system life cycle, but several tools may be brought together, if their interfaces allow for this, to provide an Integrated CASE (I-CASE) environment, embracing project management in addition to support for each phase of the system life cycle. The term Integrated Project Support Environment (IPSE) is often used in this context.

Future developments will include knowledge-based tools, i.e. expert systems, which have rules for the tasks of analysis and will be able to prompt the analyst with relevant questions.

7.7 Summary

While engaged in the investigation and development of systems, an analyst probably carries most of the necessary details in his or her head. When the analyst writes ‘order’, they know what that includes as a copy of an order is probably in front of them. However, some days or weeks later, the actual constitution of an ‘order’ itself may not reveal whether it is a company purchase order or an order for an outside customer.

The time-honoured principle of all systems work is that not all of the ramifications of the system in its finest detail can be carried in the analyst’s head. Therefore, he or she partitions, structures and records. The DD is an essential component of that record. It may start as a list on one or two sheets of paper, then expand into a set of sheets, one for each entry. Alternatively, it may be a computerised DD.

The availability of a terminal or PC greatly facilitates the development of a DD. Software that can do more than simple text processing can be a very useful tool.

Several sophisticated analysis tools which incorporate a DD are currently available, and their use is increasing rapidly. The self-checking facility of DD software for internal consistency and the ability to check and compare existing files make it much easier for the analyst to produce and maintain the DD. However, the essential criterion for a DD to be useful is that it must be used and kept up to date throughout the project, rather than being allowed to become a documentation chore late in the project life cycle.

Exercises

7.1 What is the purpose of a Data Dictionary, what is the fundamental unit defined in it and what other data objects are defined within it?

7.2 The definition of a data object requires certain information to be given about that object. What does a DD entry for a data object detail?

7.3 What are the advantages of using a Data Dictionary?

7.4 How might you implement a data dictionary in its simplest form through to more sophisticated mechanisms?