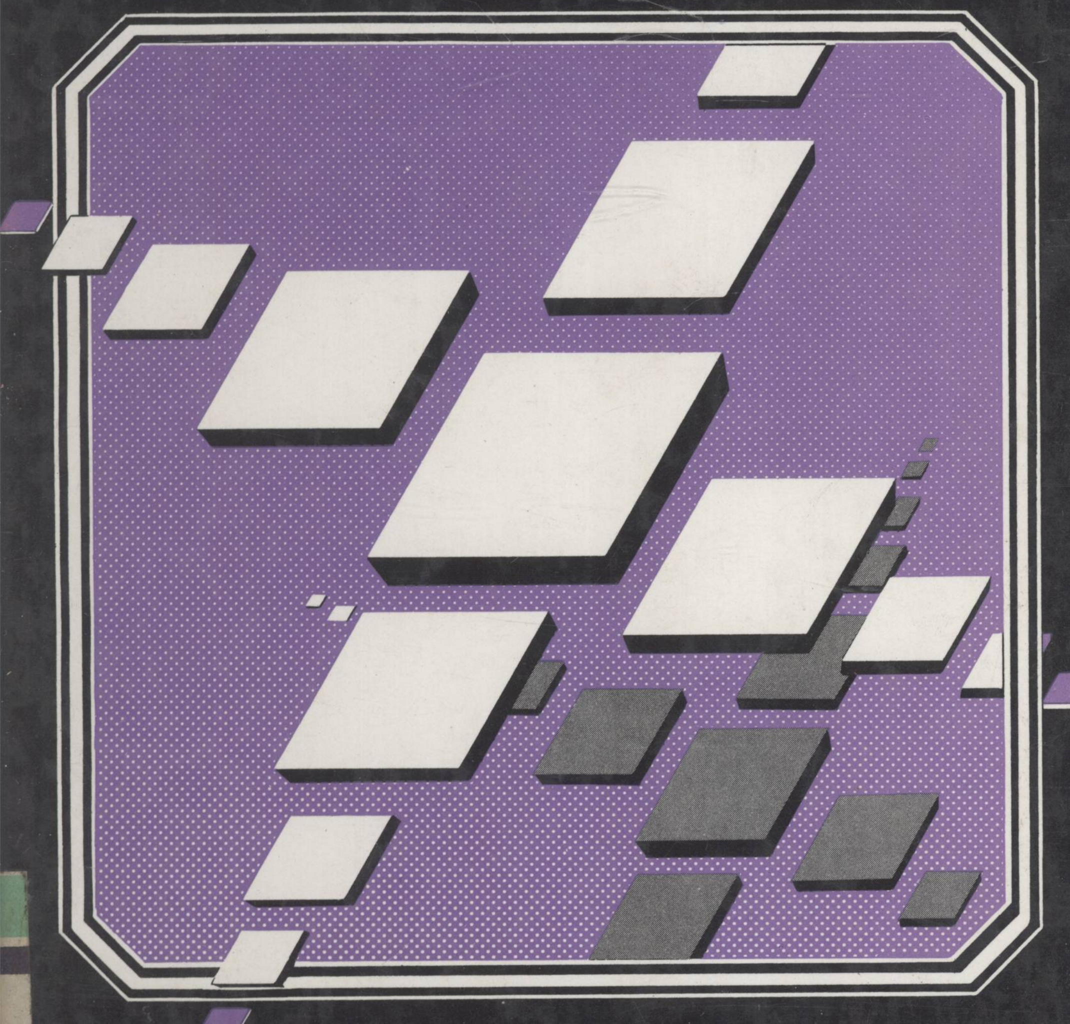
Ellis Horwood Publishers
COMPUTERS AND THEIR APPLICATIONS 12

# Systems Analysis and Design for Computer Applications

D. MILLINGTON



SYSTEMS ANALYSIS AND DESIGN
FOR COMPUTER APPLICATIONS

# SYSTEMS ANALYSIS AND DESIGN FOR COMPUTER APPLICATIONS

D. MILLINGTON, B.Sc., M.Sc.
Senior Lecturer in Computing Science
New University of Ulster
Coleraine



ELLIS HORWOOD LIMITED
Publishers · Chichester
Halsted Press: a division of
JOHN WILEY & SONS
New York · Chichester · Brisbane · Toronto

First published in 1981 by

**ELLIS HORWOOD LIMITED**
Market Cross House, Cooper Street, Chichester, West Sussex, PO19 1EB, England

*The publisher's colophon is reproduced from James Gillison's drawing of the ancient Market Cross, Chichester.*

**Distributors:**

*Australia, New Zealand, South-east Asia:*
Jacaranda-Wiley Ltd., Jacaranda Press,
JOHN WILEY & SONS INC.,
G.P.O. Box 859, Brisbane, Queensland 40001, Australia

*Canada:*
JOHN WILEY & SONS CANADA LIMITED
22 Worcester Road, Rexdale, Ontario, Canada.

*Europe, Africa:*
JOHN WILEY & SONS LIMITED
Baffins Lane, Chichester, West Sussex, England.

*North and South America and the rest of the world:*
Halsted Press: a division of
JOHN WILEY & SONS
605 Third Avenue, New York, N.Y. 10016, U.S.A.



**British Library Cataloguing in Publication Data**
Millington, D.
Systems analysis and design for computer applications.
(Ellis Horwoods series in computers and their applications)
1. System analysis
2. System engineering
3. Electronic digital computer systems
I. Title
003 Q295

**Library of Congress Cataloguing in Publication Data**
81-4140 AACR2

ISBN 0-85312-249-0 (Ellis Horwood Ltd., Publishers)
ISBN 0-470-27224-4 (Halsted Press)

Typeset in Press Roman by Ellis Horwood Ltd.
Printed in Great Britian by R. J. Acford, Chichester.

# Table of Contents

# Author's Preface

The increased use of computers in so many areas of commercial, industrial and social endeavour has lead to the rapid increase in the number of people performing "systems analysis", or "systems development" (the preferable title) in such organisations. Even more people have found themselves involved with such work as "consumers" of new systems. In universities, systems analysis is now a basic element of undergraduate courses in computer science, and also features in a range of courses for other students, particularly those of business studies. This book has grown out of notes for lecture courses given by the author to both computer science and business students. The content is not an attempt to provide a comprehensive training in systems analysis. It is directed primarily at the interface area between computing and organisation or management studies. Part 1 is a general review of the tasks and nature of systems development, and is an introduction suitable for both computing and business students. Part 2 considers a range of tools and techniques required in the work of systems development, of which some (e.g. report writing, and cost and benefit studies) may have been encountered by business students as part of other studies. A particular feature of Part 2 is the inclusion of a method for hierarchically structured systems description based on the standards of the British National Computing Centre.

As well as students in the previously mentioned categories, computer programmers wishing to move into systems analysis will find that Part 1 and Part 2 provide a sound introduction from which to proceed. The businessman involved in the introduction of a computer application will find Part 1 helps him better understand the work with which he is involved.

Some suggestions for exercises are included in the text as stimuli to thinking about the issues discussed, but there is no comprehensive set of problems. Students training in systems analysis should be taken through one or more case studies in parallel with the material in this text. Supplies of such case studies are steadily improving in published texts, and through organisations such as The Case Clearing House of Great Britain and Ireland (based at Cranfield Institute of Technology, Bedfordshire, England).

The level of knowledge of computers assumed in this text is that which might be obtained from an appreciation course, or from an introductory book, such as that by Meek and Fairthorne [1]. Neither is an essential pre-requisite as long as the reader is prepared to pause at points in the present text, and seek elucidation elsewhere for any topics and terminology which cause difficulty. For the study of the design of computer systems –files and programs– the reader is referred to books which treat these subjects in their own right, such as Waters [2] on file design, Jackson [3] on structured programming, Jenson and Tonies [4] on software engineering, Date [5] on databases and any of the multitude of books on specific programming languages.

As mentioned above this book has grown out of lecture notes and they have been developed over ten years or more. Inevitably there are many individuals, and many books, which have influenced for good the material presented. Two books which I have used extensively with students are those by Daniels and Yeates [6], and Bingham and Davies [7]. Their influence may be recognised and is acknowledged. Others, who have contributed beneficially to the material presented, include those people with whose systems I have been involved, particularly many friends in the health services – their influence is often apparent in pages of this book. Less apparent, but often important, is the contribution made by students who have suffered, or been stimulated by, my courses. I hope that many of them are now experiencing for themselves the interest that lies in the work of systems development. Discussions with, and comments from, colleagues within The University of Strathclyde, and other institutions, have likewise contributed beneficially to my understanding and presentation of the material of this book. Their influence is appreciated.

Particular value has been derived from the comments and support of the series editor, Brian Meek, director of the Computer Unit, Queen Elizabeth College, University of London.

Finally, I acknowledge with great thanks the work of those who typed the manuscript, Miss A. Wisley, Mrs. M. McDougall and Mrs. K. Millington, and the assistance of Miss S. Gemmell with the illustrations.

## REFERENCES

[1] Meek, B. L. and Fairthorne, S., *Using Computers,* Ellis Horwood, 1977.

[2] Waters, S. J., *Introduction to Computer Systems Design,* National Computing Centre Publications, 1974.

[3] Jackson, M. A., *Principles of Program Design,* Academic Press, 1975.

[4] Jensen, R. W. and Tonies, C., *Software Engineering,* Prentice-Hall, 1979.

[5] Date, C. J., *An Introduction to Database Systems,* Addison-Wesley, 1975.

[6] Daniels, A. and Yeates, D., *Basic Training in Systems Analysis,* 2nd ed., Pitman, 1972.

[7] Bingham, J. E. and Davies, G. W. P., *A Handbook of Systems Analysis,* 2nd ed., Macmillan, 1978.

# Part 1 – PRINCIPLES

CHAPTER 1

## Initial Ideas of Systems and Systems Analysis

The aim of Part 1 of this book is to provide an understanding of what is entailed in systems analysis and design for computer application. In simple terms, the question is "What do we do if we want to use the computer to assist in some area of work?" The reader with a little knowledge of computers may answer "Write a program (of instructions) for it." However, this is only part of the full answer, and it may not be the most important part. How much more is involved is the particular concern of this book. In this chapter the answer to the question raised above is considered at an overall level. Subsequent chapters are about the detailed work to be performed.

The phrases "systems analysis" and "systems design" are used technically in several professions. For example the electrical engineer uses them differently from the operations research practitioner. Even in computing differing uses are found. In the context of computer applications, systems analysis and design is concerned with organisational systems and their handling of data, but in the context of computer design and construction, the main concern is with circuitry and electronics principles. Discussion of what systems analysis is can often be at cross purposes (e.g. in a job interview) because the participants have different backgrounds, since although there is an overall level at which all systems can be considered, the differences are great at the detailed level of working. It is then important to identify *what system* is being discussed or studied.

Although there is great variety in the use – and also, one suspects, in the misuse – of the phrases "systems analysis" and "systems design", there is even greater variety in the use of the word "system". Common usage includes, for example, "a central heating system", "a system for betting on horses (or in the casino)", and "one's nervous system". Distinguishing between these particular systems presents no problems, but the systems analyst working on organisational systems may have to contend with inter-linked systems and may find difficulties in identifying their boundaries. Is there a distinction between a university's research system and its teaching system? Is the ordering of stationery in a commercial organisation a system in its own right, or just a part of the system of the department using the stationery? What is the relationship between one's betting system and one's personal finance system? The reader who attempts at this point to note down some kind of answer to any of these three questions will acquire a feel for some of the difficulties of talking about systems! It is valuable, therefore, to clarify the concept of **system**.

## 1.1 WHAT IS A SYSTEM?

Hopeman [1.1] has traced the attempts of many writers to provide a definition for the word "system", and few of those offered are at all helpful. Because the concept of a system has many features, short definitions – say, up to six lines – are either inadequate or depend on other terms which themselves require definition. However the concept of a system can be usefully established by considering the characteristics of systems.

Three systems were mentioned in the previous section – central heating, betting and nervous – and others readily come to mind. In business there are systems of production, distribution, marketing, finance, maintenance, stores, etc. Social systems include those of government, education, medicine, taxation, transport etc. There are also number systems, pulley systems, the solar system, river systems, university admissions systems, hospital appointment systems and so on.

Take any one of these systems and consider its nature. About the educational system, for example, we may note that

(i) each country has its own system
(ii) from one view, it is a large number of individual students and teachers; at different times and in different places groups of students meet with a teacher to work on a particular subject
(iii) from another angle, the educational system is a number of educational institutions – schools, colleges, universities etc. – together with area and national administration
(iv) as well as publicly-financed educational institutions, many private individuals and commercial organisations provide educational services
(v) in Britain, a child usually starts school at the age of five and continues through different schools, colleges and, possibly, universities until taking-up full-time employment; this gives a view showing a student entering the educational system at an early age and leaving it later as an "educated" (or processed?) person
(vi) the obvious intent of the educational system is to educate people.

These comments may appear to be quite random, but they do point to many of the characteristics discussed by Hart [1.2] as associated with the concept of a system.

Firstly, a system is recognisable as a whole unit. It is not uncommon to refer to the educational system of a country, but rare for one to talk of the world educational system. Most people can grasp the former idea, but few people find the latter manageable. Why is this? It is partly because of its size and partly because the educational systems of different countries do not have enough in common to justify grouping them together for any practical purpose. The concept of system requires not only that **elements** or **components** of the system be identified, but also that there are recognisable **relationships** between the components. The earlier comments (i) to (vi) on education as a system indicate several kinds of component that can be considered – students, teachers, courses, schools, colleges, universities – and the list can be extended greatly in length and detail (desks, books, etc). Not all these components are of an elementary nature. Any school can be considered as a system in its own right, and is in fact a **sub-system** of the education system, which is a **super-system** to it. The existence of sub-systems and super-systems is a basic characteristic of systems in general, and makes the work of studying systems more manageable, because many aspects of one sub-system may be considered independently of other sub-systems. In order to be able to work on a system – analysing an existing one or designing a new one – it is important that its **boundaries** be defined and that the elements and relationships included are clearly recognised, and also those which are excluded. For example, the listed comments on the educational system noted the existence of non-publicly-financed education ((iv) above) and in any discussion of the educational system of a country one must be clear as to whether private education is included or excluded; the scope of the discussion is noticeably different in the two cases.

In the list of comments about the educational system, (v) and (vi) point to further fundamental characteristics of many systems – indeed of all systems that may be studied in the context of computer application. These are that the system exists to do something – it has an **objective** – and that in pursuing its objective it processes something taken into the system, referred to as **input**, and produces something, referred to as **output**. In simple terms an educational system takes in students and produces educated people. (This statement offers scope for much discussion on the definition of terms and the specification of educational objectives, but that is best left to books in those areas!) In industry, manufacturing is a system which takes raw materials as input and produces finished goods as output. A department store has goods as input and output, and also finance as both input and output. For all large systems such as commercial organisations there are a whole range of inputs and outputs rather than just one of each. Some of these inputs and outputs – such as, for example, the ones mentioned specifically in the previous examples – will be more closely related than others to the prime objectives of the organisation.

Discussion of system objectives will occur more fully later in this chapter. At this point it is sufficient to note that a simple mechanical system may exist "to do the job" or "to do the job as cheaply as possible." More complicated systems will have a range of objectives each of which will be met partially rather than fully.

Systems having an objective and processing inputs to produce outputs are often categorised as **dynamic** systems. The study of such systems is concerned with the extent to which they meet their objectives, and the improvement of their performance is the usual aim of such studies. The introduction of computing facilities into a system may offer one way of improving performance, and this idea provides the starting point for much that comes later. However, there are still a few points about systems generally which are worth noting. Firstly, dynamic systems experience **change**, which may be internal to the system, external, or both. **Internal** changes include, for example, the occurrence of wear in a mechanical system, and changes in resources and other circumstances in organisational systems. Examples of **external** changes to a mechanical system might be a change in temperature around it, or a kick by a distraught operative. Organisational systems may experience their own kinds of crudely applied change – e.g. the shutdown of a factory–as well as more gradual changes such as an improved market for a particular product. Whatever the causes, the existence of change means that the best system today will not necessarily remain the best. It will need inspecting at regular intervals, and altering to improve its performance in changed circumstances.

As well as dynamic systems, there exist systems categorised as **static**. They have neither objectives nor inputs nor outputs. In themselves they do nothing. They exist only as frameworks, structure descriptions or sets of rules, but they may be used within larger systems which are dynamic. For example, a betting system may be formulated as a set of rules which determine how bets should be placed, and at this level is a static system. If the system boundaries are drawn more widely to include the gambler placing the bets by following the rules, the system is dynamic, has inputs and outputs (of money) and an objective (profit). Another example is given later of this change in the character of a system as the boundaries are extended. For the systems analyst and designer, static systems are of interest only as elements of dynamic systems.

Another relevant categorisation of systems is given by distinguishing dynamic systems as either **deterministic** or **probabilistic**.

For a deterministic system, the outputs for any given set of inputs are precisely defined. The structure of the system is known fully and its operation can be specified exactly. Examples of deterministic systems include gear and pulley systems and electronic devices, e.g. computers.

In a probabilistic system one or more components or relationships are known only imprecisely, or are subject to chance. For example, in the educational system the performance of a student in an examination has a wide range of variability. It may be affected by factors difficult to include within the formal boundaries of the "educational system" – the student's home circumstances, personality and state of health, and even this breakfast cereal on the day of the exam. Business and industrial systems, unless they are fully automated, are probabilistic. Any system which includes people is probabilistic in nature, and the system designer who fails to recognise this – who believes that the organisational system he designs will run like clock-work, i.e. will be deterministic – commits a fundamental error in systems design. The recognition that a system being created as a computer application is a probabilistic system, makes it particularly necessary to include, in the design work, procedures relating to monitoring and control. "Monitoring" is used here to mean identifying how well the system is functioning. Control procedures, which modify the system or its environment, will need to include some which respond to the results of monitoring. Before leaving for the present these general aspects of the work of systems design, one more comment will be made. This is, that the importance attached to the needs for monitoring and control will depend firstly on how probabilistic the system is, i.e. what is the likelihood of it malfunctioning, and secondly, on what the consequences may be when it does malfunction. For example, a delay of half-an-hour may be acceptable when producing invoices, but not when treating hospital patients.

In Figure 1.1 the ideas discussed above are related to the classification of systems which make use of computers. A computer program is a set of instructions specifying how a particular task is to be performed. It is a static system. It does nothing by itself. To achieve something it must be run on a computer. The computer and the program constitute a dynamic system which can take data as input and produce output, and the system is deterministic since the structure and mode of action of the computer with the program are precisely defined. When we consider where and how data arises in the organisation and how the results are used, the boundaries of the system are pushed out further and it becomes probabilistic in nature. This change in character of a system as one pushes the boundaries outwards occurs with many systems.

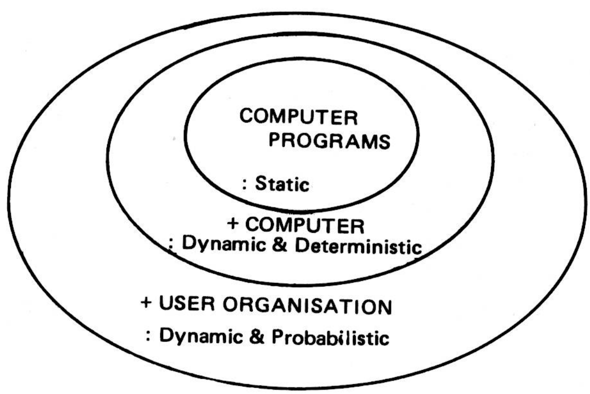



**Fig. 1.1 – Classification of computer-oriented systems.**

This section started by seeking a definition of a system, and has developed by identifying the characteristics of systems. In conclusion two definitions are quoted. The first is by Boulding in Mesarovic [1.3]

> A system is a big black box
> Of which we can't unlock the locks
> And all we can find out about
> Is what goes in and what comes out.
>
> Perceiving input–output pairs
> Related by parameters,
> Permits us, sometimes, to relate
> An input, output and a state.
>
> If this relation's good and stable
> Then to predict we may be able,
> But if this fails us – heaven forbid!
> We'll be compelled to force the lid.

A "black box" view of systems is often useful. It assumes that only what goes into or comes out of a system is important, and that if these are all right then the details of its internal working are not consequence. The reason for systems analysis and design work, i.e. "forcing the lid", is indicated by Boulding's final verse, namely the occurrence of change.

The second definition is given simply to illustrate the inadequacy of attempts to define "system" too briefly – and also shows something of the ambiguity of the English language:

"A system is an orderly mode of procedure which provides regularity of action"!

## 1.2 AN OVERALL VIEW OF SYSTEMS ANALYSIS

Problems with the terms "systems analysis" and "systems design" have been indicated previously, and it is still necessary to clarify what is meant generally by "systems analysis" in relation to the use of computers in commercial and industrial organisations. Take for the present, a simple practical view. If a **systems analyst** – defined here as the kind of person who does this kind of work – is invited to "computerise" the invoicing procedure of I. M. Bloggs' manufacturing company, what does he do? A brief answer might include the following:

- find out what is done at present
- decide on what could be done by computer and how it will be done
- introduce the computer system

This outline can be expanded into a fuller list of eight main types of task to be performed.

1. **Project definition** (or specifying terms of reference)

The area of work in which the computer may be used needs to be set down and its boundaries identified. Is it to be the solution of a particular problem, the work of one department, a function which involves several departments of the organisation, or the whole of the organisation's activities? For example, at I. M. Bloggs is it just a bottle-neck in producing invoices that is to be considered – or the whole work of handling orders – or the whole accounting function of the company – or is it that the company are considering the use of computers in all areas of work?

2. **Investigation** (or fact-finding)
Before improvement on any existing system can be attempted, it is necessary to investigate the system, to find out the facts about it. The systems analyst needs to know what is done now, or may need to be done in the future – how many invoices are now handled each day, and how many may be handled in 12 months time? He needs to find out why things are done in certain ways; why must every invoice be seen by a supervisor and what are the objectives of the system?

3. **Analysis**
Analysis of the information gathered should establish how well the present system is working. How quickly and cheaply are invoices produced? How well do security and control measures compare with general standards? Analysis seeks answers to questions such as "Does the system meet its objectives?", "What are the problem spots?" and "Where may the system be improved?"

4. **Design** (and specification)
Design work follows logically from analysis. Identifying *where* a system may be improved leads to ideas on *how* the system can be improved, and in particular, the advantages of computer use. Overall design decisions lead subsequently to detailed specifications – of computer programs, clerical procedures, documents in use, equipment required, etc. As in other spheres, the architect's design must be developed into plans for the builder.

5. **Implementation**
A new system is designed in order to be implemented, i.e. to be brought into use. It needs to be prepared, e.g. computer programs written, files set up, and, possibly, equipment and staff acquired. It needs to be tested to see if it does correctly everything it's intended to do. And it needs to be introduced to replace the existing system.

6. **Evaluation**
Evaluation is concerned with reviewing the new system after it has been in use for a while, to see how well it is working. How well have all those good intentions of the design work turned out in practice? It entails casting a look at the new system in as critical a manner as was given to the old system during analysis.

7. **Monitoring**
Evaluation is usually a once-for-all task. Monitoring is the continuing task, which is done throughout the life of the new system, that of checking how well it works. Changed circumstances may mean that the system will cease to be the best that could be used, or that it begins itself to develop problems.

8. **Maintenance** (or modification)
Maintenance is also a continuing task during the life of a system, because the need

for system modification arises often from evaluation, from monitoring and from the occurrence of the unexpected.

A list of tasks such as 1–8 is often referred to as the **system life-cycle**, although termination of a system does not appear explicitly. Also, it must be recognised that the list is only a simple statement of the kind of tasks undertaken. Although these tasks are basically in the sequence in which they are performed, in a large project such tasks may occur at more than one point in its development. For example, a project may need to be re-defined at some point, or discoveries made during analysis may indicate the need for further investigation, or even ideas occurring during design may raise more points for analysis – testing a newly-designed system may reveal important information which was not discovered earlier and makes it necessary to do some tasks again. The larger the project, or the less-throughly the tasks are performed, the greater will be the need to return to a task higher up the list. In this way the development of a new system is often an iterative process. If the need to re-tread earlier steps is discovered only after implemention, the new system may be categorised as a failure, and is likely to have been an expensive failure and expensive to rectify. In Britain, the centralised system for driver and vehicle licences, and, in the USA, the Post Office Department's Postal Source Data System (Li [1.4]) seem to have been examples of such failure.

Several further points about the work of systems analysis may be made in relation to the list of tasks 1–8. Firstly, **systems analysis** is not an appropriate title for the whole work, of which analysis is only a part. **Systems development** is a more appropriate title, and this will be used from now on. Unfortunately **systems developer** as a job title has an odd ring and is used very little. In this book the practitioner of systems development will be referred to as a **systems analyst**, which is in accord with the most common use of that title. However, the reader should note that specialist jobs and titles exist which relate to specific tasks in systems development, and that there are other titles used for the work overall, e.g. systems engineering.

Secondly, two obvious areas of skill and experience come into use in systems development. One is that related to computers and their use. The other is that relating to the system under study, e.g. invoicing, stock control, appointment bookings, etc. Knowledge of both areas is essential for successful systems development. A systems analyst may come from either background and acquire skill and experience in the other. Some companies have a strong preference for people who know their business thoroughly, while the preference of other companies is for people who know computing thoroughly. In some organisations the titles of **business analyst** and **computer analyst** are used to distinguish roles in the overall work of system development. In all organisations the systems analyst must be able to make use of the wide range of skills and experience of other staff, that may be appropriate in a project.

A third point is that the amount of work in each of the tasks will vary