Systems Analysis and Design

D.R. Jeffery

M.J. Lawrence

Systems Analysis and Design

D.R. Jeffery

Department of Information Systems
University of New South Wales

M.J. Lawrence

Department of Information Systems
University of New South Wales

PHA

Prentice-Hall of Australia Pty Ltd

Prentice-Hall of Australia Pty Ltd, Sydney
Prentice-Hall International Inc., London
Prentice-Hall Canada Inc., Toronto
Prentice-Hall of India Private Ltd, New Delhi
Prentice-Hall of Japan Inc., Tokyo
Prentice-Hall of Southeast Asia Pte. Ltd, Singapore
Whitehall Books Ltd, Wellington
Prentice-Hall Inc., Englewood Cliffs, New Jersey

Typeset by Meredith Trade Lino Pty Ltd,
Burnley, Victoria.

Printed and bound in Australia by Globe Press Pty Ltd,
Brunswick, Victoria.

2 3 4 5 88 87 86 85 84

National Library of Australia
Cataloguing-in-Publication Data

Jeffery, D. Ross (David Ross), 1948-
Systems analysis and design.

Bibliography.
Includes index.
ISBN 0 7248 1181 8.

1. Electronic data processing. 2. System analysis.
I. Lawrence, M.J. (Michael John), 1941- . II. Title.

001.6'1

U.S. ISBN 0-13-880261-0

PREFACE

While information systems have been developed and used since time immemorial, the increasing complexity of modern organizations, together with the ready availability of inexpensive computing machinery, has concentrated attention on the process of analysing and developing systems. This has led to the emergence of a body of knowledge on how the process should desirably be undertaken. This text is intended to introduce the student to this body of knowledge: the tools, techniques and methodologies for both analysing and designing an information system. The development framework presented is based on the top-down structured approach maintaining a clear distinction between the logical and physical aspects of the system.

In designing this text to fulfil the requirements of a first course in information systems it has been assumed the student will have studied, either as a pre-requisite or a co-requisite, elementary technical aspects of computer hardware and programming. Without this exposure it would be difficult for the student to appreciate the application of computer technology to an information system.

We believe this book will suit introductory systems courses in specialized programs such as computer science and in user application disciplines such as accounting, business and librarianship. It will also serve as a handbook of design procedures for practitioners.

The approach to the subject is oriented towards practicality, with emphasis on guidelines and procedures. However, we have attempted to combine the best aspects of a 'hands-on' step-by-step guide with a conceptual treatment of the issues and principles involved.

Two case studies, one batch and one on-line, are presented in the final two chapters to illustrate the application to real-life problems of the methodologies presented in the book. The instructor may wish to set parts of these case studies for reading and class discussion as the chapters on analysis and design are being taught.

We wish to acknowledge our debt to Mrs Marie Dyer who, without complaint, typed and retyped the drafts; to our colleagues in the Information Systems Department at the University of New South Wales who contributed in many ways to the content of this book; and to Miss Annmaree Wilson for the index. We also wish to express our gratitude to our wives and children — Cas, Anna and Michael, and Sarah, Stuart and Emily — for their patience and encouragement during the time of writing this book.

D.R. JEFFERY
M.J. LAWRENCE

CONTENTS

1 The Development of Information Systems

1.1 INTRODUCTION

Every organization has its own information system and makes use of many other organizations' information systems in order to carry on its activities. This is true whether the organization is a large public company or whether it is a group of people trying to achieve a common purpose, such as a club or society. The types of information systems kept and used by these organizations, however, would be very different in their nature and content. They would also be very different in the way in which they came into existence. In small organizations the systems are often quite informal and have evolved slowly over a lengthy period of time to fit the particular people involved. In larger organizations the information systems should satisfy the needs of a large number of people, necessitating a more formal approach to the definition of the system requirements and a more rigid application of the rules governing the operation of the information system. Despite their large size such systems may be developed quite quickly (6 to 12 months) using large teams of people.

An information system is defined as a means of providing information in such a way that it is of use to the recipient. A very broad view of information is taken to include not only the provision of traditional management information (decision-input information), but also the processing of data associated with the routine operation of business systems, e.g. payroll processing, debtors, etc.

In the case of a small club, a simple list of its members and the details of any annual fees outstanding might provide a very effective information system for one aspect of the club's activities. This list could provide not only the details of any financial indebtedness to the club, but also details of the members' addresses and telephone numbers which might provide the basis for all club activities. Along with the club's monthly bank statements and cheque butts, this may be the only formal system within the club. This is not to say that it is the only information that is relevant to the club, but rather the only information for which a formal mechanism may be developed for capture, storage and dissemination.

In a large public company a simple information system would be completely inadequate. The requirements of business and legislation are such that many information systems need to be developed to track the organization's activities and to provide a basis for its future activities. In one sense, the information systems of an organization are concerned with the past, in terms of providing a record of what has gone on in the organization over a period of time; the present, in terms of conveying the status of the organization at this point in time; and the future, in terms of the information that can be provided to assist in determining the best strategy for future action.

1.2 DEVELOPING SYSTEMS

The manner in which a new information system is developed depends on a number of factors, such as the size of the system, the complexity of the system, and the characteristics of the system's user group. Information systems literature contains many references to the 'user'. This term covers a very broad class of people within the organizational structure, ranging from (say) a machine operator up to the managing director. They all participate in information systems in the roles of providing data input, receiving routine data output, or making decisions based on the information supplied by a system. Consequently, the skills and knowledge base of a system's user group can vary considerably, particularly from one system to another. For this reason the systems analyst needs to be effective in communicating with a very broad range of people.

The development approach taken throughout this book follows a fairly traditional structured methodology. This approach is particularly applicable when a system is fairly large, complex, or in some way stretching the capabilities of the organization. When a task is very simple, or required very urgently, a more expedient approach may be more suitable.

1.3 THE SYSTEMS DESIGN TASK

Developing large information systems is a professional activity, requiring both judgement and creativity for a successful result. In meeting the requirements of the user, the analyst combines current technology with the skills of the user and the skills and expertise of the other members of the system's development group. This process requires considerable creativity. There are many possible ways of structuring a system, each with its own particular strengths and weaknesses. To assess the tradeoffs associated with each particular alternative, the professional judgement of the systems analyst is required.

1.4 THE TRADITIONAL LIFECYCLE

In order to provide a structure for the development of information systems, most organizations follow what is called a system lifecycle approach. This lifecycle is in many cases a list of discrete stages through which the project proceeds as it is developed:

Stage 1	Requirements specification
Stage 2	Feasibility study
Stage 3	Logical design
Stage 4	Physical design
Stage 5	Programming
Stage 6	Implementation
Stage 7	Post-implementation review

This lifecycle encompasses the two important characteristics of top-down development and structured design. These concepts are discussed in Chapter 7, but an overview would be helpful at this stage. The lifecycle and its associated methodology provide a step-by-step approach to development. At each stage of the design more detail is added until the physical specification is complete in Stage 4. A distinction is drawn in the lifecycle between logical and physical systems. A logical system is concerned with 'what do we want the system to do' and a physical system with 'how will it be done'.

Within each of the lifecycle stages there exists a set of activities to be carried out, such that the documentation might take the form of a manual for the construction of systems, not unlike a large construction project. The tasks within each stage have to be defined and completion dates set so that control over each stage and hence over the whole project can be achieved. Thus the lifecycle provides not only a framework for developing a system, but also a basis for controlling each stage of the project.

Requirements specification. This stage is concerned with reducing uncertainty surrounding the scope of work to be carried out. This usually means looking at the system's target area and gaining an understanding of its environment, identifying any pressures for change on the current system, and looking for areas in which improved performance which may be possible if new systems were implemented. The amount of effort involved in this stage will vary enormously depending on factors such as the complexity of the environment, the risk associated with a change, or the technology available to realize a particular advantage. Therefore the requirements specification task needs skills such that opportunities can be perceived, knowledge of the organization such that the environment under study can be appreciated, and knowledge of available technology such that any advantage can be gained through the use of suitable technology. These skills are very different from those required from (say) a programmer and indicate some of the skills that are needed in a team which is responsible for the development and implementation of an information system.

Feasibility study. This stage focuses on analyzing the current system and proposing alternatives which meet the requirements of the user. It is here that most of the systems analysis work is usually carried out. The aim is to explore the alternatives available to such a depth that it is possible to determine which option is the most desirable from both a technical and an economic viewpoint. Not only is an analysis of the current system carried out, but quite often some aspects of the design of the new system. It must be remembered that the feasibility study should only go as far in the analysis and design as is necessary to establish feasibility. It is a most important stage in the lifecycle because it sets the foundations for all future work and dictates the direction that the design will pursue.

Logical design. In this stage the designer sets about the task of producing a complete logical description of the new system. This entails a description of the data and data flows in the system, the inputs and outputs of the system, the control and security procedures that are necessary, and the logical operation of the system in the user environment. At this stage the analyst determines what the computer system will be doing, but does not provide all of the necessary details to completely specify how the system will operate.

Physical design. In this stage the designer fills in detail, or provides meat for the skeleton produced in the logical design. This requires the complete specification of all inputs and outputs, files or data base, security and control procedures, and the operation of the system in the user environment.

Programming. Typically the physical design must now be converted to a set of program specifications. These specifications are then given to programmers to design and code the system. A critical part of the programming task is the testing of programs both at an individual level and at the system level — where they all have to function together to provide the complete system.

Implementation. This stage aims to provide an operational system for the user. Usually this involves the installation and testing of the complete system in the user environment. Frequently the new system is run in parallel with the old in order to provide a thorough test of the new system by reconciling its output with that of the old. In other cases this parallel running is not possible and other procedures are used to verify the output of the new system.

Post-implementation review. After the system has been operating for 6 to 12 months a review may be undertaken to determine whether it is successfully meeting the requirements originally set. A review of the benefits being achieved through the use of the system may also be undertaken to provide feedback on the estimation procedures that were used earlier in the system's lifecycle.

1.5 INFORMATION SYSTEMS COMPONENTS

Figure 1–1 shows a generalized model of an information system. In this model we can identify the major components of an information system. These are:

- Data input
- Input error correction
- File update
- Reporting
- Inquiry processing

Any information system will be composed of these five basic parts, arranged in various ways to satisfy user needs.

Data input. Transaction data (e.g. orders received, products sold) are entered into the system.

Correct errors. All data input has potential for errors, and must be checked and corrected where necessary before information processing can proceed.

Update files. The transaction data (now stored in a machine readable form) are used to update the company files (e.g. stock files, employee files).

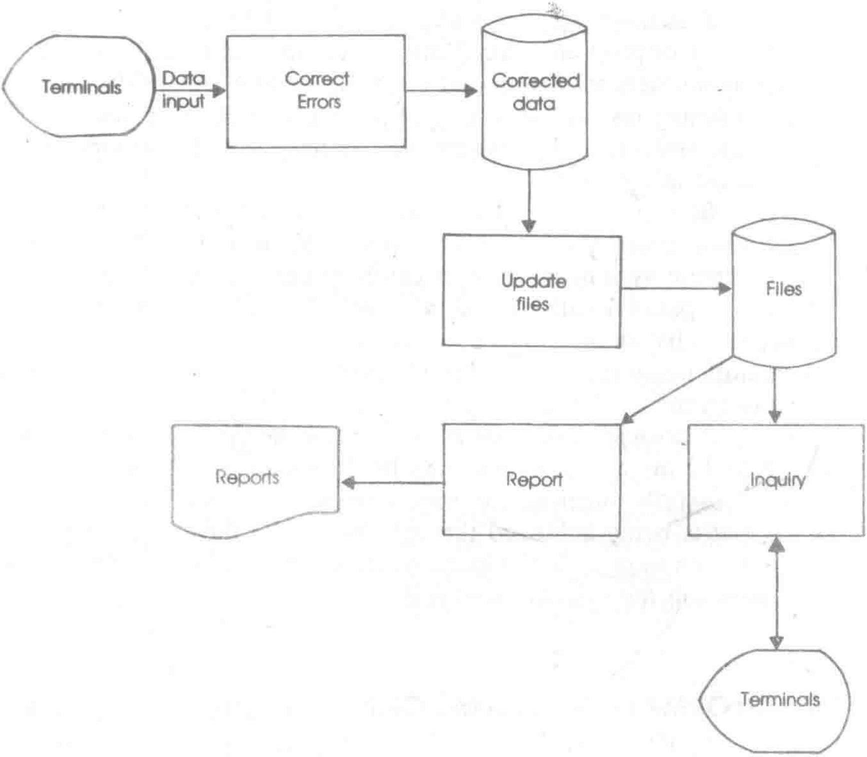

Figure 1—1 Information system model

Report and inquiry. Both of these components are concerned with accessing stored data to provide information for use in the organization. An inquiry may lead to either the printing of a report or the display of information on a visual display unit (VDU). The types of reports provided for management can be divided into four basic types:

- Scheduled reports
- Predictive reports
- Demand reports
- Exception reports

Scheduled reports can be further subdivided into those reports covering activities and those covering status. An example of an activity report would be a production report showing quantities of goods manufactured during a period by different sections of the organization. A status report, by way of contrast, might show the balance of work in process at each manufacturing point at a particular point in time.

Predictive reports show the expected or possible results of particular courses of action. A budget is one form of predictive report, while the results of a model of some part of the organization is another.

Demand reports are those which are requested by management. These can be subdivided into preformatted and *ad hoc* reports. A preformatted demand report is one in which the layout or format is predetermined and the report supplied to management when requested. An ad hoc report is one in which the format of the report is not known until the request for information is supplied by management.

Exception reports, as the name suggests, show only those items which do not satisfy a given rule. This type of reporting is very important because it concentrates management attention on important items and does not show items where no management action is necessary. An example of this is in an accounts receivable system, where management is interested only in slow payment accounts, or debts above certain values. It should be noted that scheduled reports, predictive reports and demand reports can all be prepared on an exception basis.

1.6 DATA AND PROCESS DESIGN

In the lifecycle section the development of systems was presented as a layered set of stages. Another way of looking at the development task is to consider a further sub-division of each of the lifecycle stages into three segments:

- The data segment
- The process segment
- The context segment

These three segment types are shown in Figure 1–1: the data (files, corrected data), the process (update files, correct errors, report, inquiry) and the context (people using the terminals for data input or inquiries, or people using the printed reports).

Throughout the development of a system the designer is deciding on the database or set of files which the information system will use, the processes which are carried out within the information system, and the procedures which will be used in the organization in order to use the information system. In this way the designer undertakes a database design or data analysis, a process design, and also an organizational design set of activities. These three activities take place in all stages of the analysis and design of the new system. This concept is illustated in Figure 1–2.

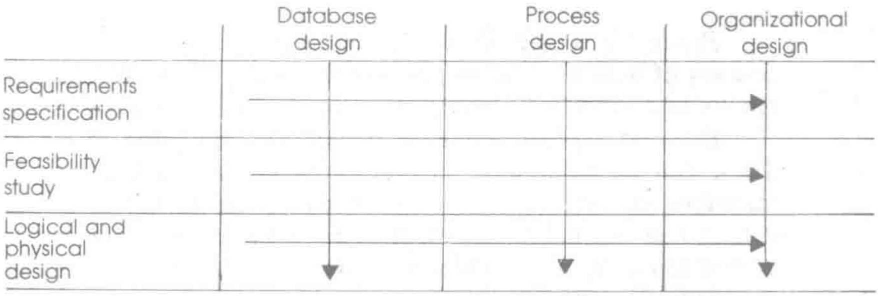

Figure 1—2 The three elements of design

1.7 ALTERNATIVE LIFECYCLES

Over the last twenty years, since the introduction of programming languages such as COBOL, BASIC and FORTRAN, there has been little change in system development productivity. This has motivated the development of:

- Very high-level languages
- Automated approaches to design
- Prototyping of systems to reduce subsequent maintenance costs

With the development of much more powerful systems software (designed to help the system builder), it has become possible to approach the development of certain classes of systems in a slightly different way. For example if a system is being developed which uses an existing database, and a less procedural (or very high level) language is available, quite often an evolutionary approach has been used to advantage. The nature of the evolutionary approach to software development is shown in Figure 1–3. After the requirements specification stage, a prototype for the system is designed and constructed, usually using the power of the very high level language. This results in a system shell which can be provided for the users to gain experience and to evaluate. As a result of this evaluation there are typically some aspects of the system's operation which are unsatisfactory to the user, and consequently modifications are identified and the prototype updated. This prototype is then evaluated again by the users until such time as a satisfactory system is established. This is not necessarily a complete system, but only a prototype which specifies how that system will look to the users. It may not contain all of the control and security procedures of the eventual system, nor (say) the full database specification.

It is still early days in the development of very high level languages and consequently in the assessment of the best systems lifecycle to use with these languages.

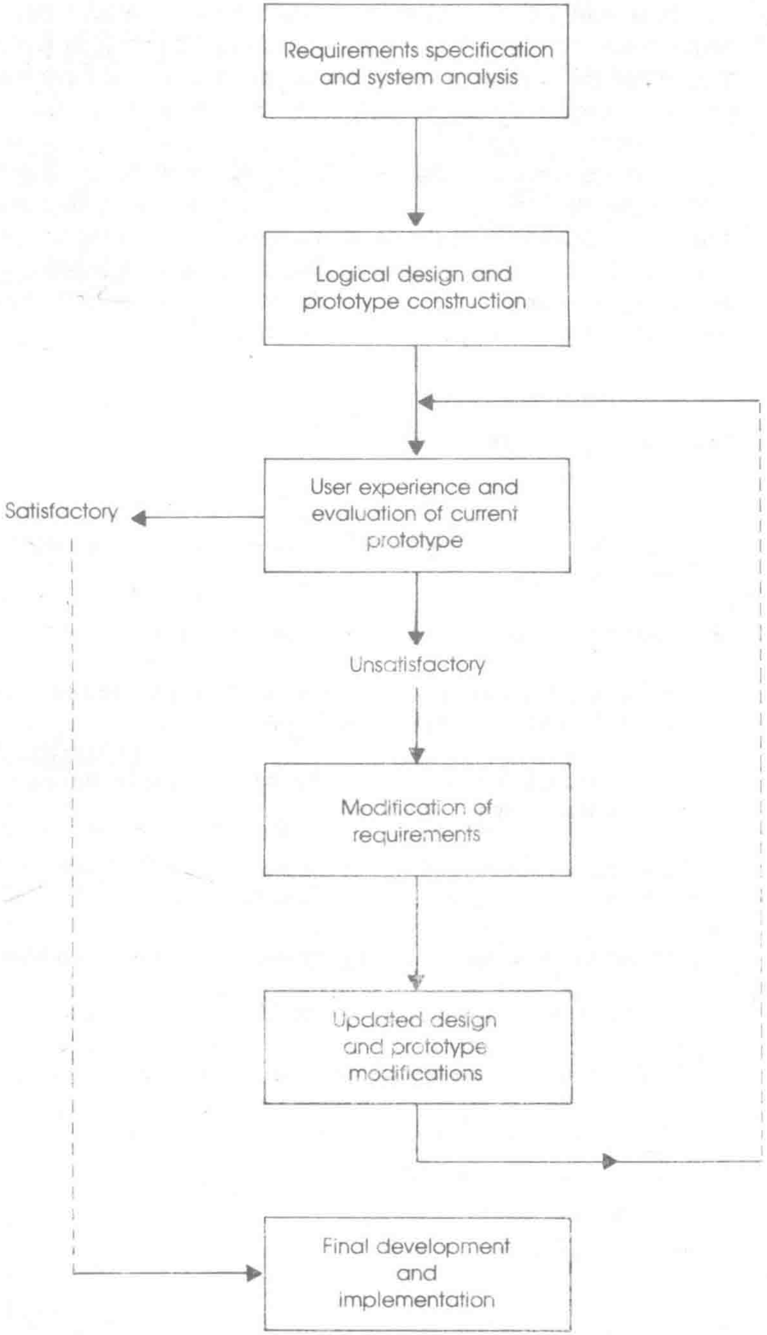
Requirements specification and system analysis
Logical design and prototype construction
User experience and evaluation of current prototype
Satisfactory
Unsatisfactory
Modification of requirements
Updated design and prototype modifications
Final development and implementation

Figure 1—3 Evolutionary software development model