Systems Analysis

Andrew Parkin

Systems Analysis

Andrew Parkin
Principal Lecturer in Systems Analysis
Leicester Polytechnic

Edward Arnold

©Andrew Parkin, 1980
First published 1980
by Edward Arnold (Publishers) Ltd,
41 Bedford Square, London WC1B 3DQ

British Library Cataloguing in Publication Data

Parkin, Andrew
Systems analysis.
1. System analysis
I. Title
003 QA402

ISBN 0-7131-2800-3

Preface

READERSHIP

This book is intended primarily for students following a taught course in systems analysis.

As a consultant in industry I have too often had the experience of teaching the principles of systems analysis to people who have been systems analysts for some time. The shortage of analysts, the urgency of business problems and the eagerness of many programmers and others to rise to new challenges have together conspired to press many a man or woman into a position of responsibility for which they were ill-prepared. Often, the results are very successful nonetheless; a testimony to the resourcefulness of the human in the face of practical problems and, sometimes, a testimony to many hours of midnight oil, unpaid overtime, weekend working, anxiety, missed holidays and neglected families. Sometimes the results are not so successful and the fruit of the analyst's labour lies abandoned, or is the cause of discontent, or requires major surgery at further cost. There is much in this book which could be useful for practising systems analysts who would like to increase their chance of getting it right next time.

SCOPE

The aim has been to produce a digestible text, tutorial in style, concentrating on essentials. The subject is a wide one and can be taken to include a vast amount of knowledge, technique, practice, folklore and theory. Three 'razors' have been applied to pare down the material: a razor of management, a razor of previous knowledge and a razor of payoff relevance.

Razor of management Project management, management of data processing departments, business planning for data processing and the conduct of feasibility studies receive only skimpy treatment. The first and last are perhaps the most contentious exclusions, since they are nearly always among the duties of senior systems analysts. My motives are purely tutorial. These topics can be taught more effectively after the student is familiar with the processes being managed or planned. All the omitted topics are presented in a companion volume, Systems Management, which is suitable for reading after this book.

Razor of previous knowledge Familiarity with computer hardware and software, programming from specification to testing, input-output devices, storage media and file organisation and access methods is assumed.

Razor of payoff relevance I have asked myself, 'Is this information, technique or skill relevant to what most analysts have to do? Does this have a high probability of being of practical benefit to a trainee analyst?' This razor, more than the others, has dictated the content and balance of the book within its target size.

Positive advice - things the analyst ought to do, how he can organise his thoughts or work - has consciously been preferred to negative advice - things which should not be done, pitfalls to avoid. This is partly because being positive is more enjoyable and partly because negative advice, once admitted, is rather limitless.

With the same intent to be positive, there is little contrast drawn between the practices suggested here and those in common use or recommended in other books. For those who want such contrasts, I would here draw attention to the early emphasis on and analytical approach to user participation; the consideration given throughout to specific participatory actions on the part of the analyst which will encourage the user towards self-help; a model of system development which does not over-simplify project phases; concern for requirements determination; the belief that data modelling is a tool which can help the analyst with any system, not just those destined for a Data Base Management System; the concern for human factors in system design and the belief that the human procedures and forms should be thoroughly checked out before programming; a methodology for file design and computer run design in which simplicity and flexibility tend to be preserved at the expense of efficiency in the first instance; a methodology which obliges the analyst consciously to review the important effects of his design rather than have some of them result by accident; the use of structured English as a program definition language; and, above all, a concern for the WHY as well as the HOW which will assist the analyst to adapt to new environments and overcome fresh problems.

ORGANISATION

The book is structured roughly as follows:

```
Explanatory text                }repeat for    }
Questions, exercises            }each section  }  repeat for
Discussion cases, assignments                  }  each
Answers to the questions                       }  chapter
References                                     }
Appendices
```

The questions, and the answers to them, are an integral part of the text. Many of the questions aim to encourage discovery of important ideas. Subsequent sections often assume that the preceding questions have been tackled and the answers read. Some of the questions are simply reinforcers, encouraging rehearsal of the text or practice of technique. Others are very challenging and call for opinion, insight and intelligence. I have not dared to label the answers anything more than 'Answer pointers' - more thoughtful answers than mine will certainly be welcome. The reader is urged to develop his answer in his own original fashion, as he has much to gain from independent consideration of the question. He will be in a stronger position to criticise my answer or analyse the strengths and weaknesses of his own. He will develop a questioning approach which is a prized attribute of the systems analyst.

The discussion cases and assignments are reduced to a bare minimum and are meant only to illustrate some of the possibilities. A greater quantity of exercise and discussion can usefully be included in a taught course. A good source of ideas for further exercises and problems is **Information Systems: Theory and Practice** by J. G. Burch and F. R. Strater (Hamilton Publishing, Santa Barbara, California, 1974). J. Race's book **Case Studies in Systems Analysis** (Macmillan, 1979) contains cases suitable for more substantial practical assignments, syndicate exercises or class discussion. I shall be glad to supply further suggestions, cases, briefs for interviewing role-play, etc. to any interested teacher.

REFERENCES

Books and journal articles by other authors have contributed at least as much to the ideas recorded here as my own direct experience. In keeping with the aims of concentrating on essentials, there has been ruthless selection of references for citation at the end of each chapter. Those from which an idea has consciously been stolen get referenced by way of acknowledgement. In addition, for each major topic, one modern source which itself cites many references has been included to aid the serious researcher. In one or two cases, further citation of specially interesting work has been too tempting. Apologies are extended to all those authors and colleagues who recognise their phrases or ideas but who do not get the acknowledgement they deserve.

TERMINOLOGY

Those who like to fasten on to buzzwords and bandy them about will find lean pickings here. A new term might help to promote a new concept, but I am more concerned with the idea than the label. Sometimes I wonder if some of my computing colleagues, having discovered the right instantly to create a word of their own choosing when programming a computer, have too readily assumed a similar right when communicating with their fellows!

It also has to be accepted that it is unrealistic to try to change accepted terminology, however good the case. A good example has been the widespread lobby to get the word 'informatics' accepted into English as a substitute for the irrational 'data processing'. Despite adoption of this word by many other European languages, 'informatics' is only limping along in academic circles in Britain and holds no sway with field practitioners. No doubt this is partly due to a cynical suspicion that it is a buzzword. No doubt it is also connected with the fact that rational PL/1 has not swept aside irrational COBOL, and for that matter the fact that we have not all thrown away our mother tongues and taken to Esperanto. People not only tolerate, but actually enjoy, idiosyncrasy and irrationality. They are likely to change their practices only if there is some good and urgent reason.

In the original draft of this book, there was adopted (more by accident than by design) the phrase 'formal message system'. The idea of a formal message is very important to the subject of data analysis and I believe experienced systems analysts use the concept even though they may not articulate it. Clinging on to this phrase, I spoke of formal message systems which communicate those messages which have a predefined content. The phrase covered paperwork clerically prepared, as well as computerised systems. The term 'data processing' is often used loosely to include such applications as message switching or word processing where the messages are in natural language and not predefined. These are outside the scope of this book. Thus it seemed to me that 'formal message systems' preserved the generality I intended and was also more precise. However, taking the lesson of 'informatics', 'formal message systems' has been systematically replaced with 'data processing systems'.

COMPLETENESS

Nearly every chapter could have been the subject of a book in itself, but I did not set out to write an encyclopaedia or to cover every possibility. Although several sections have lists of points, the lists are not intended to be exhaustive of the possibilities nor comprehensive checklists for practical work. An attempt at completeness can be found in the book **System Development Methodology**, by G. F. Hice, W. S. Turner and L. F. Cashwell, (North-Holland/American Elsevier, revised edition, 1978). The checklist approach adopted there will surely benefit the practising analyst who wants to be methodical in his consideration of alternatives and attention to detail.

ACKNOWLEDGEMENTS

I am particularly indebted to The National Computing Centre, Manchester, England, for granting permission to reproduce their standard forms from their publication Data Processing Documentation Standards. Likewise, I am grateful to The British Computer Society for permission to reproduce the society's Code of Practice and Code of Conduct.

A large number of people have contributed specific ideas, criticisms and corrections. I would like publicly to thank Dr. K. D. Eason and Dr. D. R. Howe for their help.

Anyone with a family who has tried writing a book will know that it is not only the author who bears the strain. My 'family' includes students who get a bit neglected when I have the bit between my teeth. My thanks to them for their forbearance. Most of all, though, I am grateful to my wife, Valerie, and children, Emma and Vicki, for their help and encouragement.

Leicester
1979

A.P.

Contents

INTRODUCTION

ANALYSIS

BUSINESS SYSTEM DESIGN

COMPUTER SYSTEM DESIGN

IMPLEMENTATION

APPENDICES

1 Introduction to systems analysis

1.1 WHAT IS SYSTEMS ANALYSIS?

Systems analysis, for the purpose of this book, is defined by what a systems analyst does. A reasonáble definition of what a systems analyst does could be as follows. In connection with a proposed computer-based data processing system, the systems analyst:

1 conducts a study of the feasibility of the system;
2 liaises with users of the system and determines their requirements;
3 finds out the facts important to the design of the proposed system;
4 determines the human and computer procedures that will make up the system, designing forms and files;
5 writes program specifications;
6 tests the programs and the system;
7 participates in the implementation of the new system;
8 documents the system.

One might add a practical note:

9 turns his hand to anything else within his competence which will further the organisation's desire for an effective and efficient system.

Systems analysis is an extremely difficult job to do well but it is very stimulating. There has been a consistent world-wide shortage of good systems analysts. Good systems analysts can earn very high salaries.

Questions

(Answer these as rationally as you can before looking at the answer pointers. The expected maximum time you should spend in thinking out and writing down the answers is given in brackets - you may choose to give up if you much exceed this. Answer pointers are given on page 6.)

1 'Teleology' is the name given to the practice of assigning human purpose to something which is non-human. For example, 'the roots of the plant seek out water' or 'the steam tries to equalise pressure by escaping through the valve'. Is there any phrase in section 1.1 above which strikes you as faintly teleological? (2 min)

2 An analyst/programmer is a person who is employed to do both systems analysis and programming. Some organisations have a policy of employing analyst/programmers as it makes for flexibility in allocating available staff to the required work. What effect do you think this policy has on (a) the quality of the analysis done? (b) the quality of the programming done? (c) staff recruitment and training? (5 min)

3 Which job is the most satisfying? (a) Systems analyst; (b) analyst/programmer; (c) programmer. (2 min)

4 An orthogonal list of classifications is one in which the items on the list are mutually exclusive, the boundaries between items being clear, without gaps, so that the list covers the complete subject. What deficiencies are there in the orthogonality of the list of 'what a systems analyst does'

above? (5 min)

1.2 THE PARTICIPANTS IN SYSTEM DEVELOPMENT

Overall responsibility for running an organisation is in the hands of 'top management'. This term might include a managing director and other working directors, executive officers, and executive committees. Of course, top management are answerable to boards, councils, courts, senates or to other groups representing shareholders, electors, taxpayers, and so on. Top management could be defined as the managers who are responsible for reaching agreement on plans with other power-holders of the organisation and for directing the everyday actions of the people within the organisation in such a way that these plans are carried out. In all but the smallest organisations this usually entails dividing the work among departments, each of which has a manager responsible for a function or objective defined by top management, e.g. sales, production, research and development, accounting, data processing. Large departments may be further subdivided into smaller sections or units. The departmental and unit managers usually have authority which is circumscribed by their superiors: they must do this, they may not do that, they must refer certain decisions for higher authority but can decide for themselves on others.

The activities of the data processing department nearly always have, or potentially have, sufficient importance for the organisation to warrant the attention of top management. Even when data processing costs are a small part of an organisation's total costs, the consequences for the organisation may be great. Top management should be concerned with:

getting the maximum benefit from data processing by ensuring that worthwhile opportunities are identified and successfully taken up;
assessing the sensitivity of the organisational achievements to breakdown, error or other disruption of data processing operations, taking steps to ensure that threats are suitably contained.

In organisations where departmental managers are allowed, or required, to make most of their decisions without referring to higher authority, top management should be concerned with:

making statements of policy to guide departmental managers in their decisions;
establishing some method which will optimise the way in which data processing resources are allocated among departments;
establishing policies or rules which the manager of the data processing department should follow when dealing with competing service requests from departmental managers.

Where departmental managers normally have less discretion in executing their duties, or where it is decided that the managers will not enjoy their usual freedom specifically in connection with new data processing projects, top management should be concerned with:

identifying opportunities for computerisation;
making plans for implementing the systems and assigning priorities to the systems;
defining responsibilities for implementation of the systems and monitoring their development.

Top management may decide to increase the participation in this decision-taking by delegating their powers to a committee comprising representatives of themselves and other interested parties, e.g. managers of the affected departments, the data processing department, staff whose work may be affected. Such a committee is usually known as a steering committee.

A department affected by a proposed system is often called (by systems analysts) a 'user department'. The manager of such a department is a 'user manager', and the people in the affected department(s) are called 'users'. This last is a dubious term for two reasons. First, the term may embrace several groups in the department with different interests at stake - for example, people who prepare and handle data which is input to the system, people who receive output from the system directing their actions, people who get information from the system and use it to support their decisions, people who do not use the system directly but are affected by the decisions it supports or by the procedures it entails. Secondly, people who use the system or are otherwise affected by it may be outside the department most affected - people in other departments, the external auditors, customers, suppliers or even the public at large. Systems analysts sometimes include the people outside the affected department in the term 'user', sometimes not: some cynics have suggested they should be called 'victims' instead! Even without cynicism, it can be seen that some of the people called users are in fact servants of the system. Sometimes the term 'user' is applied loosely to embrace top management as well: in fact, anyone outside the data processing department.

This term 'user' has too much currency for an alternative seriously to be advocated. However, it is important to identify what it means in context and to make sure it does not obscure something important. It could be misleading to group together such varied classes of people under the single term 'user'.

The remaining participants in systems development are the systems analysts, programmers and operators who are usually employed in a central data processing department. They are controlled by a data processing manager, who assigns them to work on a given proposed system.

Questions

1 Systems analysts have been called 'agents of change'. An Agent is a person who represents a Principal. The Agent is supposed to further the interest of the Principal, in as much as the Principal has given him explicit or implied authority to do so, and he is supposed to refer to the Principal for a decision on matters outside his authority. An Agent's position is unsatisfactory if he represents more than one Principal and they have conflicting interests. In relation to a particular proposed system, whose Agent is the systems analyst? In other words, who is the systems analyst's Principal, to whom he should refer for a final decision on, say, the details of the system requirements? (5 min)

2 Should the analyst have authority to
a) decide the colour of the new bills sent to customers?
b) decide the best layout of a new report to be sent to a number of managers?
c) decide which people in a department should record new data required by the system?
d) decide how much the people affected by the system should participate in the design of it?
e) decide what data is to be held on the computer's files? (5 min)

1.3 PARTICIPATION BY USERS

The development of a new system is normally carried out by a team of analysts, programmers and users, although sometimes there are no users. The team is managed by a project leader.

User participation in system development could be classified as:

a) 'insider participation', in which the participant is project leader or

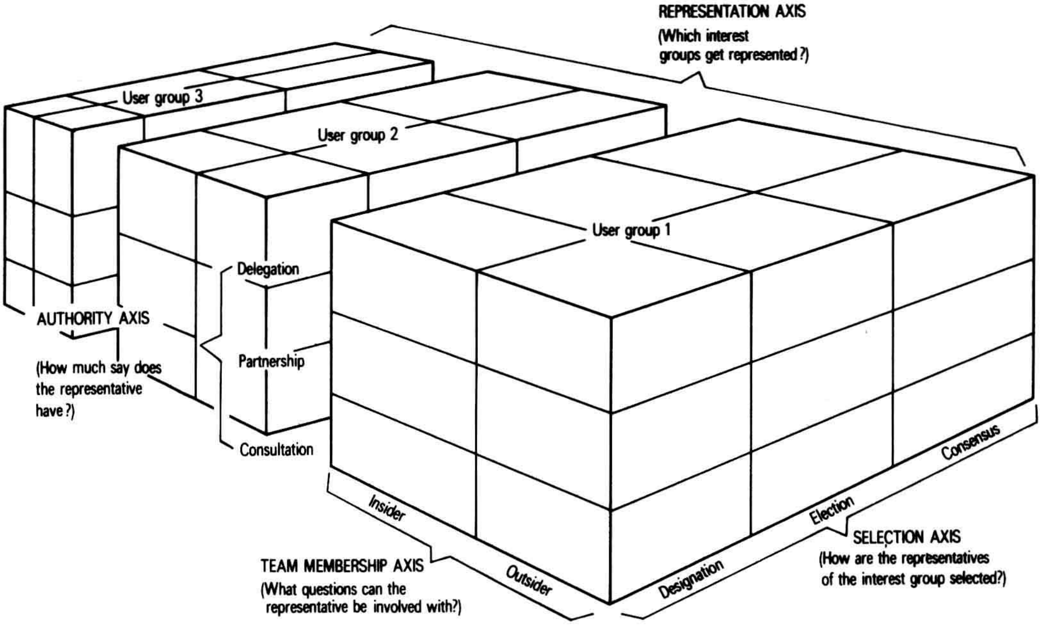
REPRESENTATION AXIS
(Which interest groups get represented?)
User group 3
User group 2
User group 1
Delegation
Partnership
Consultation
AUTHORITY AXIS
(How much say does the representative have?)
Insider
Outsider
TEAM MEMBERSHIP AXIS
(What questions can the representative be involved with?)
Designation
Election
Consensus
SELECTION AXIS
(How are the representatives of the interest group selected?)

Fig. 1.1: Dimensions of participation

another member of the project team;

b) 'outsider participation', in which the participant is not a member of the project team, but has some other opportunity to influence system development.

Insiders are usually privy to all design decisions, so they may have some opportunity to influence any aspect of the system design should they wish to do so.

User participants may be **designated** by the sponsor or by a subordinate acting on the sponsor's authority (the sponsor might designate himself), they may be **elected** as a representative of a user group, or they may comprise an entire user group, a **consensus.**

Outsider user participants may be involved through **consultation, partnership** or **delegation.** In consultation, the project team analyses the system and produces a planned design. Then the opinions or preferences of users are sought and these are taken into account as far as possible. Although consulted participants may have no authority over the system design, in practice it is usually difficult for a project team to ignore their opinions or preferences without a convincing reason. Sometimes a consulted participant may have power of veto.

In partnership, the project team describe the problems they see to the users (and vice versa) and the two groups work together to produce a joint design of the new system or some aspect of it. In delegation, a user group is given some portion of the system to design for themselves, and they can exercise autonomous choice over the features of its design.

Insider participants are nearly always involved through partnership or delegation, although it is not impossible for an insider to have a purely consultative role.

If participation does not take place, users will probably be **told** to accept a new system designed by the project team (perhaps told indirectly, by being presented with a fait accompli) or they may be **sold** on the new design. Selling may take the form of persuasion to accept the new system, either by stressing the benefits to the users or by offering compensation so that the cooperation of the users is, in effect, bought.

Although these dimensions of participation have been presented in neat compartments, in practice they are more like points in a spectrum of possibilities (see Figure 1.1).

Questions

1 It has been said that user participation is desirable because it will lead to a more **effective** system. Do you think this is likely to be so? Can you think of arguments for and against? (20 min)

2 It has been said that user participation is desirable because users have a **right** to influence the system design. Does such a right exist? (2 min)

3 Do you think users have a **responsibility** to participate in the system design? (5 min)

DISCUSSION CASES

1 The management of a medium-sized factory were considering an improvement to their system for maintaining control over their stocks of raw materials. The factory had a friendly atmosphere and all the workers called the plant manager 'Dad'. It was clear from the outset that the new system would entail small changes to the way the production workers recorded their output at the end of the day, and more substantial changes to the procedure followed by the storeman who was in charge of raw materials. The systems analyst assigned to the job got on friendly terms with the production workers, the storeman and the production department managers. He involved all these parties through partnership, using the production foreman as the representative of the production workers.

When the analyst presented his description of the proposed system, about six months after starting, it contained no surprises to anyone and he was sure there would be no objections - or, if there were, they would be very minor points which could be easily fixed. It was his opinion that although the involvement procedure had been time-consuming and required more effort, this was more than offset by the increased effectiveness of the system and the acceptance of it by everyone concerned. The managers were completely satisfied with his efforts.

2 A systems analyst was assigned to implement a new payroll system in a plastics plant, during a period of high unemployment. The comptroller, who supervised the three clerks responsible for preparing the payslips for the plant workers, had already decided that the new system was to be a standard one run by a local computer service bureau. The purpose of the new system was mainly to reduce mistakes, to increase reliability and to improve information available to management. The existing system was overstretched.

When the systems analyst interviewed the payroll clerks, it was clear that they were very apprehensive about how the new system would affect their jobs. The systems analyst implored the comptroller to explain to them company policy about the new system. The analyst also wanted to consult the payroll clerks, or involve them as partners, with regard to the part of the system which affected them, but the comptroller declined, on the grounds that it was nothing to do with them. The system was implemented about two months after the analyst was assigned. Shortly before it became operational, two of the three payroll clerks resigned, mainly because of their apprehension over the new system and their resentment at what they considered the high-handed attitude of the comptroller. The comptroller was easily able to fill the vacancies. New clerks were trained and in office with virtually no disturbance to the system. The comptroller was very pleased with the system when it was operational.

ANSWER POINTERS

Section 1.1

1 If the term 'organisation' in item 9 of the list describes an organised system of people, plant, premises, materials, money, etc., then the phrase 'organisation's desire' is somewhat unhappy. Presumably it is only the people who have desires.

2 There is nothing in the rules to say that an analyst/programmer should not both analyse as well as a systems analyst and program as well as a programmer. There are people who would perform well in either capacity. Against this, systems analysis and programming are both demanding jobs. It takes an exceptional person to be a good programmer and an exceptional person to be a good analyst. It may be considered that it would take an even more exceptional person to stay good at both. Experience suggests that analyst/programmers are generally **either** good at analysis **or** good at programming - generally the latter. This may have more to do with the difficulty of maintaining training and practice across such a broad front than with innate personal characteristics.

A reasonable answer to the question is therefore that the price paid for flexibility in allocating staff is **either** some reduction in the quality of the analysis **or** some reduction in the quality of the programming **or** increased difficulty in finding or training the desired calibre of personnel - or some combination of the foregoing.

It would be wrong to leave the impression that the only justification for analyst/programmers would be increased ease of assigning people to jobs. It may also be a good way to train a programmer in systems analysis. It may allow programmers to take on some of the analysis work which is more concerned with the computer, leaving the systems analyst more free to concentrate on the business side of the system. It may provide increased satisfaction for a particular programmer who needs a more demanding job. It may help involve an analyst in the implementation of his end product.

3 If job satisfaction depends on the extent to which a job meets the job-holder's aspirations, expectations and needs, assertions about satisfaction can be made only in relation to a particular person or, possibly, a class of person with known characteristics; this question does not make sense since it implies that satisfaction is independent of the people concerned.

4 The more obvious ones are as follows: the 'requirements of the users' (item 2) are 'facts important to the design' (item 3), so this suggests some overlap; item 7 could contain all the others unless the word 'implementation' were narrowly defined; item 9 is phrased in catchnet fashion, picking up anything that slips through the others, so if it has any substance the others must be deficient in covering the complete subject. Item 9 is also somewhat tautological, since it is coming close to saying 'a systems analyst does ... anything else that a systems analyst does'. It may also be considered that 'conducting a study' (item 1) could entail 'liaising with users' (item 2)

and 'finding out facts important to the design' (item 3), perhaps more.

There is not necessarily anything wrong with classifications that are not orthogonal (it all depends on the use to which they are put) but increased orthogonality often increases usefulness or understanding. Perfect orthogonality is hard to achieve; perhaps impossible when the thing being classified is a real-world system, as opposed to an abstract one. The concept of an ideal orthogonal division is important in systems analysis and will arise several times later in the book.

Section 1.2

1 The Principal, it could be proposed, should be someone with the authority to make decisions about matters which involve conflicting interests of different user groups. He probably authorised the systems analysis and design in the first place. Thus, where departments are autonomous, and where a departmental manager has commissioned a system, the department manager is the Principal. Otherwise, the Principal will be the top manager who authorised the work, or the steering committee if it exists and if it authorised the work. Although a steering committee is not an individual and comprises people whose interests may conflict, the end product of debate in the committee is that it speaks with a single voice to outsiders, so is acceptable as a Principal.

I have in the past called this person/body the Commissioner and have also heard the term Prime Client used in the same connection. One author calls this person the User, which is rather confusing. At least we are agreed that he is an important participant who should be identified by the systems analyst, even if there is no widely accepted term. Possibly the most apt term, which is gaining currency, is 'Sponsor'; this is the term which will be used for the rest of this text.

Perhaps it should be pointed out that just because someone has called systems analysts 'agents of change', it does not follow that they are and that they must have Principals. The idea of a systems analyst exercising personal authority over conflicts of interest, deciding what is best and answerable to no-one save his own conscience, has a certain whimsical appeal, like the Lone Ranger; but he is unlikely to find many organisations where this view of the power he should have is shared by others.

2 There is nothing to stop top management delegating authority to whoever they think fit, although an exception may occur with some responsibilities which are defined by law. If they believe that the systems analyst would be the best person to make decisions on the matters listed, they should give him the authority. In the absence of express authority, though, each of the decisions listed - colour of bills, report layout, personnel assignment, participation in decision-taking, what data is kept on record - will be the responsibility of some other person or persons in the organisation, probably the managers of the affected departments. Assuming that top management continue to have confidence in those other persons, the systems analyst should not have authority to make the decisions listed. So my answer to each part of the question is 'probably not'. However, this does not mean that the systems analyst may not have an opinion on these matters, nor that he should not volunteer advice to the decision-taker nor, indeed, that he cannot try to persuade the decision-taker if he believes a poor decision is being taken.

Section 1.3

1 Users who have not participated will be told or sold. A told or sold user is more likely to react negatively to a new system than one who has participated to the point where he is confident the system will satisfactorily cater for his needs and preferences. A negative reaction by a user may

take the form of hostility to the change, lack of cooperation, or diminished enthusiasm. If this results in less effectiveness, then lack of participation is likely to lead to reduced effectiveness. If the user has participated to the point where he identifies the new system as being the result of his own choice, then he is less likely to react negatively when it is in operation.

Users have expert local knowledge, i.e. knowledge of facts which are important to that part of the system with which they are concerned. They may because of this be capable of making a more effective design which caters for local conditions, exceptions, preferences and interactions, ensures that local group social needs are satisfied, takes advantage of local opportunities and avoids local risks or pitfalls. A systems analyst is unlikely to acquire equivalent knowledge within a reasonable time.

Any system of which men are a part depends for its effectiveness on the performance of those men. Systems analysts sometimes fall into the trap of supposing that the human part of the system will operate mechanistically in a predetermined manner. Particularly when the human job is routine or low-skilled work, there is a tendency to assume that 'anyone can understand how that job is done', or to assume that user reactions are 'obvious'. This is probably a mistake; humans are complex and it is very difficult to predict the reactions or performance of men and women in a complicated system such as a data processing system. Even a simple data processing system involving humans must be considered complicated compared with a mechanical system. In practice, the only reliable indicators of the acceptability, ease of use or satisfactoriness of the system in operation are likely to come from those who will be operating it. This is a strong argument for user participation, at least at the level of consultation about the aspects of the system which concern them, or for having the users take part in experimentation of the proposed system, so that their reactions can be established.

Users without suitable aptitude, training and experience, may be unable to design a system which is as technically effective as that designed by a systems analyst. They may make poor choices on matters that do not directly affect them - for example, the timeliness, accuracy, reliability or security of data passed outside their group. They may miss technical opportunities through lack of knowledge of possibilities. They may place wrong emphasis on the importance of different parts of the system with which they are concerned. They may design a system which is open to disruption or other risk in circumstances which are not presently prevailing but which might be foreseen by an analyst. They may fall into pitfalls familiar to the analyst.

The more extensive the representation, the more selection tends towards consensus and the more there is insider participation, so the greater are the difficulties of organising and coordinating participation, the more likely it is there will be conflict over how the system should operate, the more debate there is likely to be in attempts to reconcile conflicts and the longer the time scale for implementation, possibly resulting in lost benefits. The grand purpose of a new system may be lost if the design responsibility is distributed far and wide. If conflicts remain, after having been exposed and sharpened by debate among the representatives the eventual choice may cause more discontent, or the choice will be postponed for fear of the eventual consequences.

My answer to the question 'Is participation effective?' is 'Yes, usually, but exactly what sort of participation are you talking about?' The most effective form of participation will be one which fits the particular system, the particular people and the particular priorities of the moment.

2 The reason only two minutes were suggested for this question is that I thought either you would have strong views on this subject and would deliver an immediate opinion, or you come from a country like Norway or