Information Systems Development Research: An Exploration of Ideas in Practice

D. E. AVISON[1] AND A. T. WOOD-HARPER[2]

[1] *Department of Computer Science and Applied Mathematics, Aston University, Birmingham, B4 7ET, UK.*

[2] *Department of Mathematics and Computer Science, University of Salford, Salford, M5 4WT, UK.*

This paper describes aspects of applied research related to the development of information systems. There is a proliferation of information systems development methodologies and some confusion has resulted. Methodologies are classified into six broad themes, but experience suggests that no methodology can be appropriate to all situations. A contingency framework called Multiview has been devised which includes descriptions of relevant techniques and tools. The analysts and users select those aspects of the approach which are appropriate to the context, in effect creating a unique methodology for each application. The approach has been used in a number of applications and these experiences have led to modifications of the framework. In the cycle of action research there is close interaction between theory and practice and between the researcher and the practitioner. The paper describes the learning process and discusses some lessons learned from this experience. It also puts forward conclusions about the methodology framework, the strengths and problems of using a contingency approach, and of action research in information systems.

Received November 1990

1. INTRODUCTION

The work described in this paper is outlined in Figure 1 which has eight parts. There has been a proliferation of information systems development methodologies (1) over some years and this has led to practitioners, researchers and teachers being confused about which methodology is 'best'. A significant part of our work (and that of many others) has therefore been comparing and classifying information systems development methodologies (2), and in Section 2 we discuss six broad themes to information systems development.

However, we argue that it is unreasonable to rely on one approach. Each of the themes has strengths and weaknesses and our practical work suggests that tools and techniques appropriate for one set of circumstances may not be appropriate for others. The appropriate methodology will depend on the context (3), that is, the organisation itself and the users and analysts who are developing the information system. As argued in Section 3, it is not feasible for analysts to know sufficiently well many methodologies, one which is chosen for each problem situation. On the other hand, it is also unreasonable for analysts and users to choose techniques and tools without any methodology framework to guide their choice. There are too many possible alternatives and too many decision points. A compromise is to use an approach where the choice of techniques and tools can be made within a loose methodology or framework (4).

In Section 4, we describe a contingency framework, called Multiview (5), which is partly a synthesis of many of the themes discussed in Section 2 including modifications of techniques and tools (6). An important aspect of this research has been to use Multiview in a number of problem situations (7). In Section 5 we discuss some lessons learned from the experience of using Multiview. These experiences have led to modifications being made to the framework. We then discuss some conclusions about the methodology framework, the use of such a contingency approach and of action research in information systems (8).

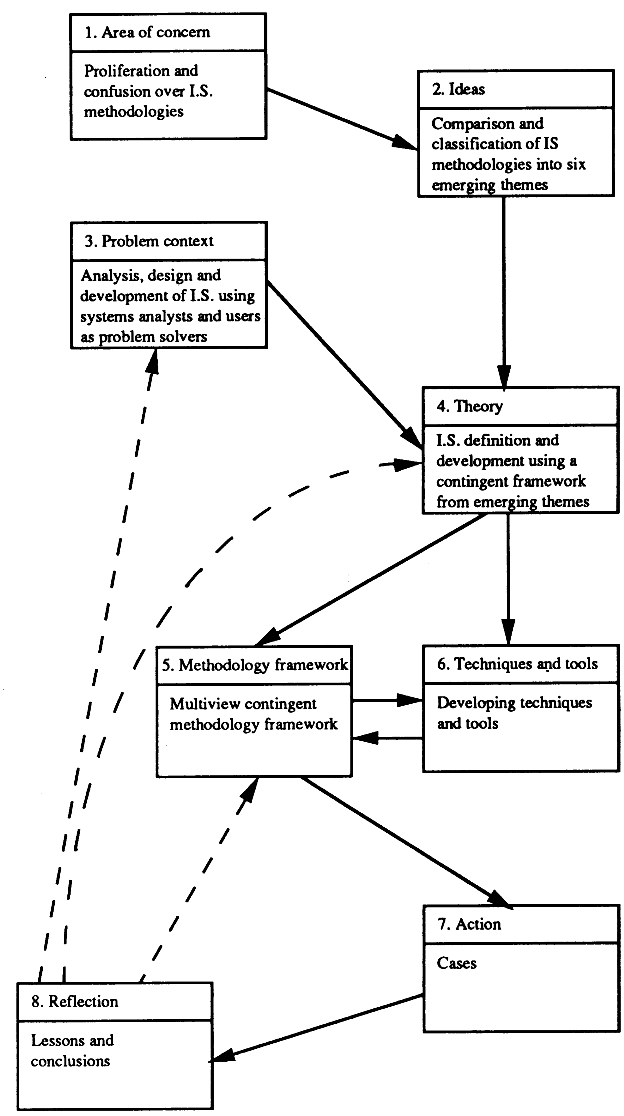

Fig. 1. Learning cycle of action research

This paper stresses two aspects of this work: first, the place of Multiview in the world of information systems development approaches and second, the representation of this work as a learning cycle. The framework is presented as a structure to help the users and analysts choose those documentation aids, procedures, techniques and tools, from those suggested in Multiview, which are appropriate to the particular problem situation. Some choice within the framework is enabled in the context of the application – some 'freedom in a rocking boat'.[88] Thus Multiview is not seen as a recipe but a paradigm for interpreting information systems development in context.

In reviewing the learning cycle, we discuss learning about methodologies for developing information systems, learning about Multiview itself, and learning about the use of Multiview in different situations. We provide first a review of information systems development methodologies.

2. THEMES IN INFORMATION SYSTEMS DEVELOPMENT METHODOLOGIES

There appears to exist many hundreds of system development methodologies.[57] Bubenko[20] argues that:

> It is a reasonable estimate that hundreds of more or less similar methodologies have been published. In practice, probably tens of thousands of more or less different approaches are being used. Most organizations have developed their own methodology and prescribed it in the organization's (data processing) handbook.

The number of approaches reflects different viewpoints, cultures and experiences, though, as Bubenko suggests, many methodologies are similar. There have been many attempts either to compare methodologies or to provide frameworks for their comparison.[4,18,31,38,40,47,60,73–77,90,96]

Avison & Fitzgerald[4] identify six broad themes. Each theme attempts to address one or more of the weaknesses of the conventional approach, such as the NCC methodology.[32] This approach and its variants are still being used in a number of data processing departments. Its weaknesses include:[2]

- The failure to meet the needs of management.
- Unambitious systems design.
- The inherent instability of models of processes.
- Inflexibility due to the output driven design.
- User dissatisfaction.
- Problems with documentation.
- Maintenance workload.
- Application backlog.

Perhaps the problems of the conventional approach are inherent in its philosophy which, when crudely stated, might be that 'computerised data processing systems will lead to gains in reduced cost and increased speed of processing'. These benefits have not always been apparent and, in any case, give a narrow perception about what an information system is attempting to achieve. The alternative themes stress different aspects of the systems development process, suggest different techniques and tools and, more importantly, are based on different philosophies.

The *systems approach* stems in modern times from the work of Bertalanffy.[17] Although often considered to be too impractical and wide ranging,[96] many of its principles have been taken up by the information systems community, and are an integral feature of a number of information systems methodologies, in particular soft systems methodology[26,27] and viable systems diagnosis.[14,39] Its principles would suggest that attempting to develop information systems for the widest possible context would be advantageous, as developing an information system for a small part of the organisation might be to the detriment of the organisation as a whole. It highlights the importance of the relationship between an organisation and its environment, and in multi-disciplinary teams to understand organisations. Soft systems methodology offers help in understanding large and complex problem situations.

Planning approaches can involve strategic management in information systems work so that their needs are analysed and that information systems are implemented which do more than computerise the operations level applications. Like the systems approach, planning approaches attempt to identify the needs of management, and plans the ways of meeting these needs. This helps to ensure that systems are not implemented in a piecemeal fashion and that there is overall planning for information systems development. Three approaches which address these issues more fully are BIAIT,[23,25] ends/means analysis[35,94] and IBM's Business Systems Planning (BSP).[49,62]

In *participative* approaches all users are expected to contribute to and gain from any information system, and this should increase the likelihood of its success.[41,52,65] Effective Technical and Human Implementation of Computer-based Systems (ETHICS)[66] is a participative systems design methodology and, as well as being an acronym, the name is meant to imply that it is a methodology that embodies an ethical position. As well as being a methodology based on the philosophy of the participative approach, it encompasses the socio-technical view, that for a system to be effective the technology must fit closely with the social and organisational factors. This should help to prevent technically viable information systems failing because of 'people problems'. Other methodologies, though not based on participation in the same way as ETHICS, could be termed people-oriented (rather than process or data oriented). For example, ISAC[58,59] seeks to identify the fundamental causes of users' problems. The approach is designed to analyse users' problems and to solve aspects of them where appropriate.

The theme of automating some aspects of the information systems development process is an abiding one, and has intrigued developers for many years though early attempts at automation have not been very successful due to the technology available.[43,44,48,85,86,93,102] *Prototyping* has been enabled by the availability of software tools, in particular fourth generation systems, which greatly reduce the costs associated with prototyping. A number of sources[13,19,21,46,55,56,64,71,82] evidence the interest in prototyping. Prototyping enables users to comment on the proposed information system, its inputs, processing and outputs, before the system has been designed in its final form and therefore, like participation, reduce the likelihood of user dissatisfaction. It avoids the inflexibility of the output driven design of the conventional approach.

Further, through the use of prototyping tools, the application backlog can be reduced by the greater speed of information systems development.

Structured approaches aid the understanding of a complex problem through functional decomposition and the associated documentation techniques. Explanations of the approach[36, 42, 69, 70, 92 103, 104] tend to stress techniques such as decision trees, decision tables, data flow diagrams, data structure diagrams, and structured English, and tools such as data dictionaries. The techniques used should reduce the maintenance burden and enable greater control of the development process.

Data analysis is a useful modelling technique and the data model produced is likely to be relevant for a longer period than models of processes which can be unstable. It is argued that data analysis has proved successful in creating a model which is independent of any database system, accurate, unambiguous and complete enough for most applications and users. Its success comes in the systematic way by which it identifies the data in organisations and, more particularly, the relationships between these data elements, the 'data structure'. It is also an important stage in the creation of a computer database[2, 80] and this can increase the viability of data sharing and thereby reduce the likelihood of inconsistency and inflexibility of information systems.

The more popular methodologies of today, such as SSADM,[37, 72] Merise[79] and Information Engineering,[63] are essentially blends of the data and structured approaches and do, in our view, follow in the tradition of the conventional approach. Although they use more up to date techniques, they have many of the problems of that approach. Users are beginning to discover their weaknesses, describing them, for example, as being based on 'yesterday's philosophy'.[89]

The six themes discussed above are very different and yet none of the alternative approaches to information systems development has proved to be a panacea, many having had little impact on the methods used by systems analysts. Indeed it could be argued that they are inferior to the conventional approach in some respects, in that they do not all stress some of the important aspects of that approach, for example, thorough and detailed documentation standards for communication, training and so on.

Further, each of these approaches has been attacked on a number of fronts.[5] For instance, *systems theory* has been attacked on the grounds that it represents an ideal academics' position and is not relevant to the practitioner. It expands the problem boundaries too far, leading to revolutionary change which is not appropriate for most organisations. Further, systems approaches do not describe methods for implementing solutions. They might help to understand problem situations, but they do not provide a scheme for solving problems. *Planning approaches* do not in themselves address the need for participation of all users. Further, they tend to be mechanical and flexible following the identification of strategic needs. *Participative approaches* might lead to inefficient systems designed by those who are good managers, clerks, or salesmen, but poor, and unwilling, systems analysts. It has been the authors' experience that managers have sometimes refused to participate fully, arguing that the consultant was the systems expert and should be designing good information systems for them.

Many analysts using *prototyping* methods concern themselves with the user interface and do not address the fundamental problems of systems analysis. They are simply making poor systems palatable to users. For example, rearranging headings and columns of a report may give a more aesthetically pleasing report layout containing data which still does not help the decision maker. Other possible problems of prototyping include the relative times necessary to construct a prototype and that for the operational system. In breaking down a system into manageable units and then even more manageable units, *structured analysis* may offer a simplistic view of a complex system and fail to identify fully the importance of the links between subsystems. Finally, *data analysis* may not solve the underlying problems that the organisation might have. Indeed it may have captured in the data model the existing problems of the firm, and make them even more difficult to solve in the future.

It is therefore unreasonable perhaps to rely on a methodology falling into any one of these themes. As well as the criticisms mentioned above, there are other arguments which would suggest that one approach can never be the full answer.[3]

- The tools and techniques appropriate for one set of circumstances may not be appropriate for others.
- The 'fuzziness' of some applications require an attack on a number of fronts. This exploration may lead to an understanding of the problem area and hence lead to a reasonable solution.
- As an information system project develops, it takes on very different perspectives or 'views' and any methodology adopted should incorporate these views, which may be organisational, technical, economic, and so on.

Many authors and practitioners have argued that a more flexible approach to information systems development is likely to be a more useful approach than a 'one best way' model which can lead to 'elaborate and bureaucratic methodologies'.[16]

3. CONTINGENCY APPROACHES

Davis[34] advocates the contingency approach to information systems development where the methodology chosen will depend on the particular circumstances where it is to be applied. In other words, different methodologies will be used for different situations, a kind of 'horse-for-courses' approach. He suggests that the level of uncertainty in a situation is critical in this choice, and argues that four components of the overall system affect the total level of uncertainty:

- The complexity and ill-structuredness of the system.
- Its current state of flux.
- The number of users affected, skills needed and skills possessed.
- The level of experience and skill of the analysts.

The methodology adopted therefore will be contingent on the particular situation, according to the complexity of the problem and the levels of user and analyst competence. For example, in situations of high uncertainty, a prototyping approach might be adopted. Other writers[22, 24, 38, 53, 61] have either extended Davis's work on uncertainty or offered further factors which will affect methodology choice.

The implication here is that organisations will avoid standardising on one chosen methodology because there will be circumstances where it is not particularly suitable. For each particular situation they will have available a number of different information systems development methodologies from which they choose one.

Iivari[50] presents a different contingency framework which emphasises contingent approaches within the methodology rather than between methodologies. This is the type of contingency approach discussed in this paper. Iivari suggests that the choice of tools and techniques used in an application following a contingency framework will depend on:

- The comprehensiveness and depth of the information systems design process required.
- Whether the designers choose a 'goal-oriented' strategy or an 'alternative-oriented' strategy. The goal-oriented strategy negotiates on what is to be achieved, and then proceeds to find ways to accomplish the tasks. The alternative-oriented strategy does not assume that consensus can be reached on the goals, but rather that negotiation must occur on how we are doing things.
- The choice of an appropriate adaptation strategy, reflecting on the perception about future events. One choice is to ignore future requirements, the second is to assume they are predictable, and the third is to assume that they are unpredictable, but can be dealt with.
- The choice of an appropriate implementation strategy.[100]

Multiview is a contingency approach providing a flexible framework (following Iivari) as an alternative to choosing between different methodologies or standardising on one particular methodology. The techniques and tools available within the Multiview framework are chosen and adjusted according to the particular problem situation. It is a blended methodology drawing on aspects of each of the six themes discussed in Section 2. The soft systems analysis of human activities[26, 27, 95] and the participative and socio-technical views[52, 67] has been wedded to the more conventional work on data analysis[81, 83] and structured analysis[36, 42] so as to create a theoretical framework for tackling computer systems design which attempts to take account of the different points of view of all the people involved in using a computer system.

However, these earlier works have been interpreted in a particular way (indeed, the original authors may not agree with the interpretation) and since the publication of the original text on Multiview[99] other work has been assimilated into the Multiview framework found in Avison & Wood-Harper.[9]

There are theoretical and academic antecedents to Multiview, as well as those based on commercial experience. But we need to test out our theories. To evaluate a methodology, and in particular one based on contingency, it is necessary to see it working in different situations and being used by different analysts. It is also important that the process of analysis and its results should be looked at from different viewpoints. What looks good to the analysts may not be so impressive to the user. Multiview has been used in a number of problem situations and this experience serves as commentary on the theory of analysis expounded. In other words, they address the questions: 'does Multiview work in practice?', 'what problems accrue from using such an approach?' and therefore 'how can it be improved?'. The experiences from the many real-world applications of Multiview are used as a basis for the final section which discusses the lessons gained and the conclusions made.

In none of the cases can we say that 'classic examples of how systems analysis should be done' have been provided. In a practical discipline one must always distinguish between the 'ideal methodology' as taught in text books, and the realities of any situation which causes departure from the ideal in order to allow for the exigencies of the real world. The applications expose the difference between what one would like to find in an ideal world and what is in the real world. Many design methodologies are prescriptive not only of what must be done but of the order in which it has to be done. This is not always feasible in the real world.

Multiview is an explorative structure, a loose framework with which to define and develop information systems. Multiview incorporates many of the principles, techniques and tools of a number of the themes discussed into a blended approach or meta-methodology. It is also a contingent approach, in that alternative techniques and tools are available which may be chosen according to the dictates and requirements of each situation. The user of Multiview in effect produces from this structure a unique method for every project.

Some authors have argued for a 'tool kit approach' to systems development but without a framework such as that which Multiview provides. The flexibility provided by a 'cut and paste' approach to developing information systems has many advantages.[16] But there are many problems with the tool kit approach, some of which are readily identified by management services people in the real world. In a reply to this paper, it is suggested that the 'DIY analysts' who followed this approach would be akin to the 'tradition and common sense analysts' that operated before information systems development methodologies were adopted, producing idiosyncratic and unmaintainable systems of variable value.[6] The tools themselves are useful, and they might be appropriate in different situations, but choosing which tools are appropriate (and when they are appropriate) is a very skilled job, and those with these skills are few and far between.

A reflective practitioner, who may not want to follow a rigid methodology, still needs to use a coherent approach (see also Checkland & Scholes[27] for a discussion of methodologies in practice). The Multiview approach is a contingency approach offering alternatives, but it has been designed so that the techniques and tools hang together within a framework. This is described in the next section.

4. AN OVERVIEW OF MULTIVIEW

As we have seen, the chase for the perfect methodology is somewhat illusory, because different methodologies represent different views of the world. Information systems design could be seen as a logical, technical or people problem. Different analysts have adopted different methods because they have taken a different view of the situation. There are also differences in the systems

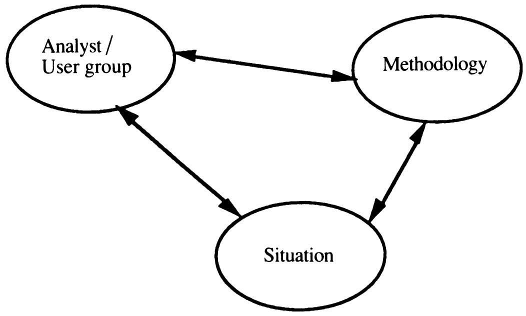

Fig. 2. The interaction between the analyst/user group, the problem situation and methodology

analysis and design approaches used which are caused by differences in the situation in which the analyst is working. Approaches that may be successful in a large bureaucratic organisation may well be different to those which work in a small fast moving company.

There exists in actual systems analysis and design practice a three-way relationship between the analyst/user group, the methodology and the situation, shown in Figure 2, but parts of the relationship are missing in many expositions of information systems development methodologies. For example, many methodologies assume – though the assumption is not stated – that each situation is essentially the same and that analysts are similar in background and experience.[33]

This section describes an approach to information systems development which combines important aspects of some of the major methodologies and themes discussed into a coherent and yet flexible approach. The methodology[3] covers five different stages of systems analysis and design, each with its own appropriate view of the problem, and each with methods for tackling that aspect of the problem.

The stages of the Multiview methodology and the inter-relationships between them are shown in Figure 3. The boxes refer to the analysis stages and the circles to the design stages. The arrows between them describe the inter-relationships. Some of the outputs of one stage will be used in a following stage. The dotted arrows show other major outputs. The five stages are:

4.1. Analysis of human activity

This stage in Multiview concerns the search for views of the organisations. The rich picture represents a subjective and objective perception of the problem situation in diagrammatic and pictorial form, showing the structures of the processes and their relationship to each other. It can be used to identify problem themes: conflicts, an absence of communication lines, shortages of supply, and so on. Through debate within the organisation, it is possible to identify relevant systems which may relieve problem themes. The root definition is a description of the major relevant system on which to focus attention. The root definition is then analysed to ensure that all necessary elements have been identified and included: that is, the Owner of the system, the Actors involved, the Client, the Transformation that takes place, the Environment in which it takes place, and the world view (or

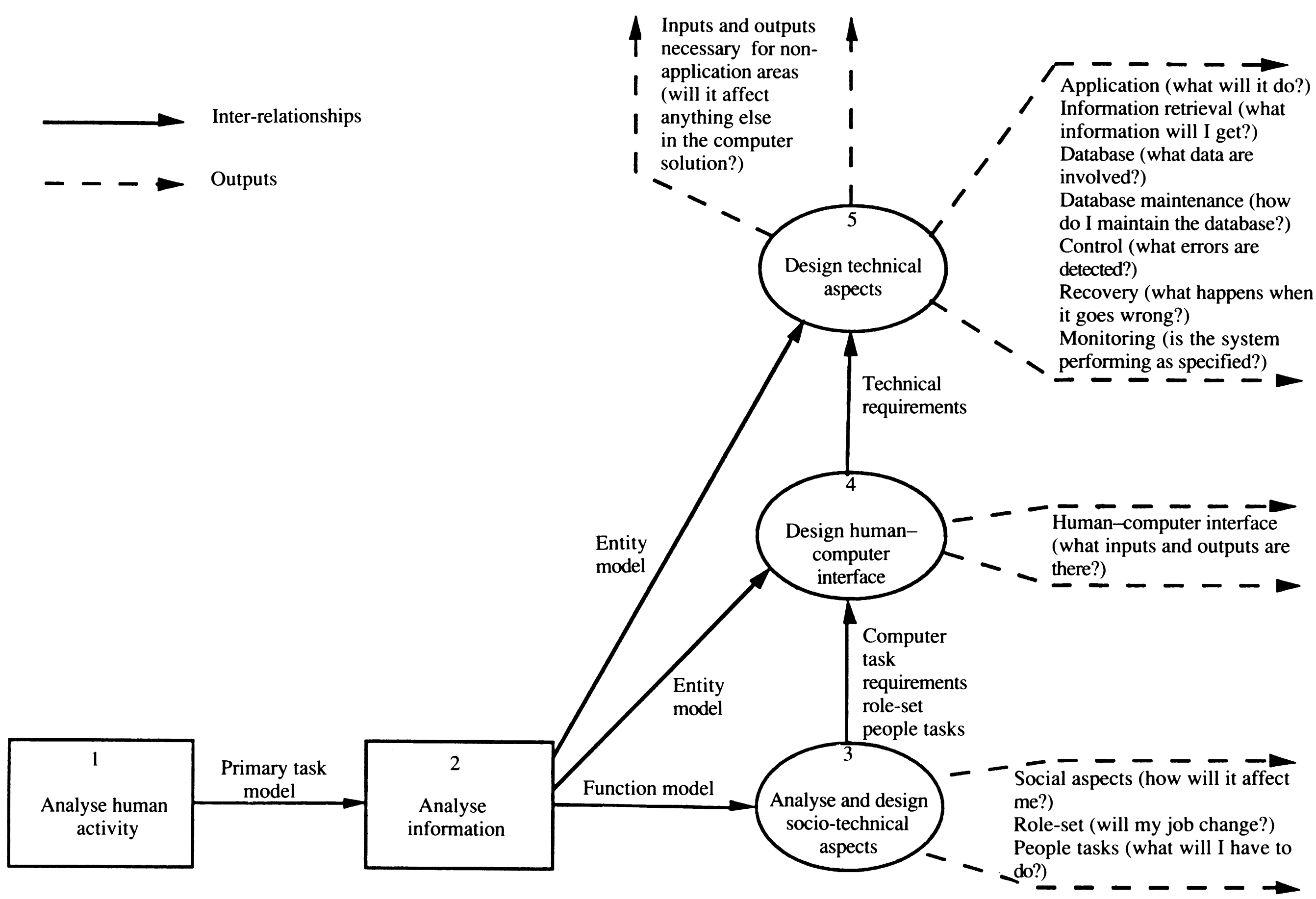

Fig. 3. The Multiview framework

Weltanschauung) assumed in our root definition. Changing the order, this check-list is called the CATWOE criteria. The activities necessary in the system defined in the root definition are also identified and compared to the representation of the 'real world' in the rich picture. In some cases the output of this stage is an improved human activity system and the information systems development process stops at that point.[27]

4.2. Analysis of information (sometimes called information modelling)

The purpose of this stage is to analyse the entities and functions of the system described, independent of any consideration of how the application system will eventually develop. By using functional decomposition, it is possible to break down progressively the main function (clear in a well-formed root definition) into sub-functions, and, by using data flow diagrams, the sequence of events and data flows are analysed. In developing an entity model, the problem solver extracts and names entities (anything you want to keep records about); relationships between entities; and the attributes which describe the entities.

4.3. Analysis and design of socio-technical aspects

The task at this stage is to produce a design from an analysis of people and their needs and the working environment on the one hand, and the organisational structure, computers, and the necessary work tasks on the other. Thus social and technical objectives are set and alternatives specified and compared so that the best socio-technical solution can be selected and the corresponding computer tasks, role sets and people tasks can be defined. The emphasis at this stage is not on development, but on a statement of alternatives, according to important social and technical considerations.

4.4. Design of the human-computer interface

Decisions are made at this stage on the technical design of the human-computer interface, for example batch or on-line and menu, command or form-driven. Specific conversations and interactions are then designed. The users are expected to be the major contributors of this stage. The technical requirements to fulfil these human-computer interfaces can then be designed.

4.5. Design of technical aspects

Using the entity model (from stage 2) and the technical requirements (from stage 4), a more technical view can be taken by the analyst because human considerations are already both integrated with the forthcoming technical considerations. This technical design will include the application subsystems and the 'non-application subsystems'. These are the information retrieval requirements subsystem, the control subsystem, the database, the database maintenance subsystem, the recovery subsystem and the maintenance subsystem.

The five stages incorporate five different views which are appropriate to the progressive development of an analysis and design project, covering all aspects required to answer the vital questions of users. These five views are necessary to form a system which is complete in both technical and human terms. The outputs of the methodology are shown as dotted arrows in Figure 3.

Because it is a multi-view approach, it covers computer-related questions and also matters relating to people and business functions. It is part issue-related and part task-related. An issue-related question is: 'What do we hope to achieve for the company as a result of installing a computer?' A task-related question is: 'What jobs is the computer going to have to do?'

The distinction between issue and task is important because it is too easy to concentrate on tasks when computerising, and to overlook important issues which need to be resolved. Too often, issues are ignored in the rush to 'computerise'. But, you cannot solve a problem until you know what the problem is! Issue-related aspects, in particular those occurring at stage 1 of Multiview, are concerned with debate on the definition of system requirements in the broadest sense, that is 'what real world problems is the systems to solve?' On the other hand, task-related aspects, in particular stages 2–5, work towards forming the system that has been defined with appropriate emphasis on complete technical and human views. The system, once created, is not just a computer system, it is also composed of people performing jobs.

One of the difficulties of representing Multiview diagrammatically is to show the basic framework and to suggest that for any particular problem situation it can be adapted according to context. Figure 3 should therefore be seen as one interpretation of the approach which is not followed in all cases, for example, that of the district health authority.[9] Within the context of problem situation, framework and user/analyst group, a unique methodology will be derived (using the term methodology here to mean 'a study of methods').[26]

Another representation of the methodology, even more simplistic, but useful in providing an overview for discussion, is shown in Figure 4. Working from the middle outwards we see a widening of focus and an increase in understanding the problem situation and its related technical and human characteristics and needs. Working from the outside in, we see an increasing concentration of focus, an increase in structure and the progressive development of an information system. Each stage addresses one of the following five questions posed:

1. How is the information system supposed to further the aims of the organisation using it?
2. How can it be fitted into the working lives of the people in the organisation who are going to use it?
3. How can the individuals concerned best relate to the computer in terms of operating it and using the output from it?
4. What information processing function is the system to perform?
5. What is the technical specification of a system that will come close enough to doing the things that you have written down in the answers to the other four questions?

Most computer scientists, programmers and technical analysts are primarily interested in question 5. It is their interest in the technology that probably brought them into computing in the first place. Most people working on large systems have developed a great interest in question 4. This is because untangling the logic of all the

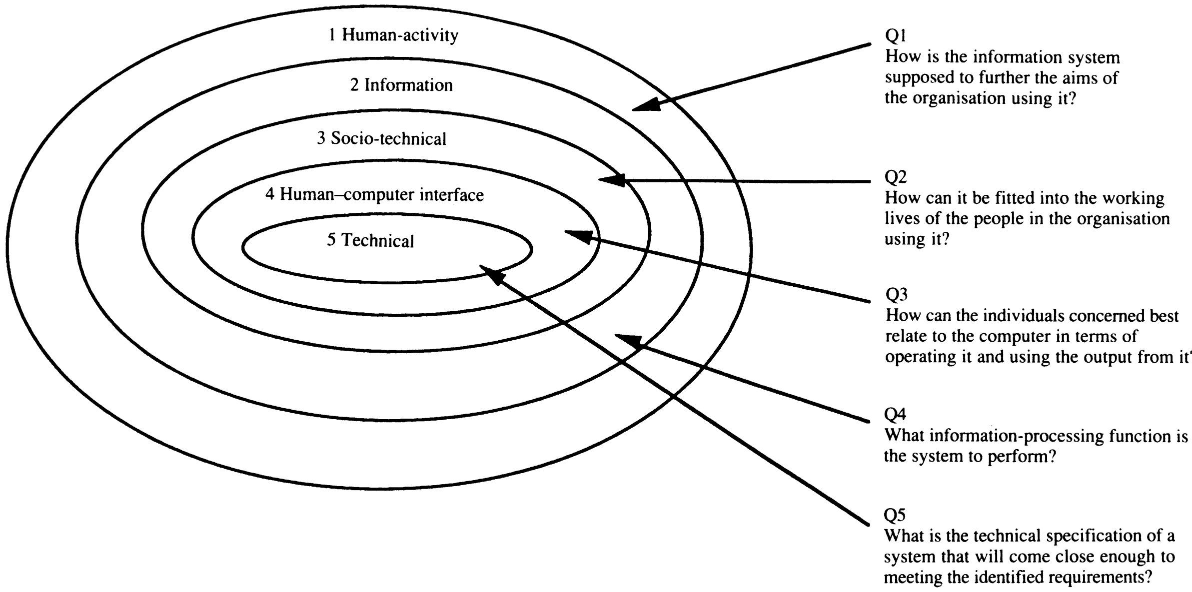

Fig. 4. The Multiview methodology

different records that need to be kept provides considerable intellectual satisfaction.

The interest of computer people in these two questions is reflected in the fact that there are large numbers of books on the technical aspects of computer hardware and programming, and an increasing number about the techniques of information modelling, that is determining the flow of information through the system and the way it has to be processed and stored. Most teaching in academic computer science departments has centred around these two aspects of information systems design, and it may be entirely appropriate for computer scientists to specialise in these areas. However, good answers to these two questions are not sufficient to ensure successful information systems.

Question 3 relates to the human-computer interface. Multiview looks at this aspect of information systems at length. Trade unions have taken an interest in question 2, even though British and American unions have been behind their Scandinavian counterparts (who have had legislation to back them). Question 1 is what the managing director has been asking all along. Unfortunately too many senior managers have had trouble with the answers. Either they did not understand the jargon in which proposals from would-be suppliers were couched or bitter experience convinced them that the answers contained more sales talk than substance. All too often managers themselves have not had the right training or advice on how to harness information technology to the needs of their operations.

Information systems are often seen as technical systems which have behavioural and social consequences. This is a rather narrow view. They are better seen as social systems which rely to an increasing extent on information technology. The technology is only a component. This wider perspective of information systems requires that a number of stakeholders in an information systems project are identified. These will include the individuals or group who request the new system and authorise the work to go ahead; the managers, who are responsible for the organizational functions in which the system is embedded; clerical staff and others, who operate the system; computer staff, who create and maintain the system; and external users of the system, such as customers or claimants, who receive the computer-produced invoices and forms. One major objective of Multiview is to help them to communicate with each other about what is needed and what is proposed. This may not be easy, because they may be looking at the system in different ways. One group may think in economic terms, another in terms of status and responsibilities, another in terms of job satisfaction, and yet another in terms of the nature of contact with the organisation.

5. EXPERIENCES USING MULTIVIEW

In this section we discuss some lessons coming from experiences in using Multiview in a number of real problem situations. Six applications are described as cases in Avison & Wood-Harper[6] and these cases form the basis for the observations and conclusions made in this section, though other cases are found elsewhere.[11,101]

Lesson 1: a blended, contingency approach can work in practice. Multiview is a blended approach to information systems development. It has incorporated aspects of the themes explored earlier: systems approach, planning approach, participation, prototyping, structured approach and data analysis. It has been argued that it was unreasonable for an organisation to rely on any methodology falling into any one of these themes for all information systems development. Furthermore, it was also argued that one rigid approach to information systems development can never be a full answer, because the tools and techniques appropriate for one set of circumstances may not be appropriate for others; the 'fuzziness' of some applications require an attack on a number of fronts; and as an information systems project develops, it takes on very different perspectives or 'views'. Multiview is an explorative structure, a loose framework with which to define and develop information systems, a contingent approach, in that alternative techniques and tools are available which may be chosen according to the dictates and requirements of each situation. However, Multiview does have a framework guiding the user and therefore avoiding a completely loose 'tool kit' approach.

The many applications of Multiview, including the six cases described in the main text,[9] evidence that it could work. Multiview has been used in small, medium and large departments, for package and tailor-made solutions on micro, mini, and mainframe computers, representing large as well as comparatively small investments, and in situations where the roles of analysts and users varied. In each case a prototype was developed which has proved useful. Sometimes the prototype, once modified to the users' satisfaction, 'became' the working system. In other applications the prototype was used as a basis for the information system developed, the prototype being used as part of an investigation phase.

Lesson 2: an information systems development methodology is complex and difficult to learn. The wide range and large number of techniques and tools that need to be described in an information systems development methodology make such a description long and complex and therefore difficult to learn and to master. Multiview is, perhaps, even more difficult because alternative techniques and tools are included. Further, it does not follow the usual rigid step-by-step description with deliverables, well defined, at each step, because of it contingent philosophy.

In all the applications, the actors using Multiview for the first time took some time to understand the methodology and therefore to use it in the application. The original exposition of Multiview[99] was not thorough and this caused problems for the actors when attempting to use it, as evidenced in the particular case described in that text. The descriptions of some techniques, for example, were inadequate and therefore they were difficult to use.

We have also found that our students only fully understand it following use of the methodology in an action learning situation, in other words using the methodology 'for real'. Without using the methodology in some context, the students do not understand the methodology. The action learning method of teaching information systems is advocated amongst other methods. Experiences of teaching and learning Multiview can be found elsewhere.[1, 8, 10, 97, 98, 101]

Lesson 3: the conventional descriptions of information systems methodologies are inappropriate. This methodology, as evidenced by the field work, does not, in practice, exhibit the step-by-step, top-down nature of conventional models and none of the applications have exactly followed the methodology as espoused in this paper.[9,54] The users of the methodology will almost certainly find that they will carry out a series of iterations which are not shown in the model. Further, in the real-world cases undertaken, some phases of the methodology were omitted and others were carried out in a different sequence from that expected. For example, in one application, some of the decisions relating to the technical subsystem were made very early in the project. In other applications, aspects of the methodology were interpreted and selected depending on the context. The appropriateness of the use of some of the techniques also varied. For example, the comparative value of conceptual models, data flow diagrams, normalisation, entity life cycles, structured English, structure diagrams, decisions trees, decision tables and entity models varied greatly between applications and even within an application as preferences for various representational forms differs from individual to individual. The usefulness (and availability) of various tools such as database management systems, data dictionaries, fourth generation systems and project control tools also varied between applications.

An exception to this rule proved to be rich pictures. These are not as familiar to most analysts and users as many of the other techniques and tools listed above, and yet they were well liked by users and managers even though they were in all cases a new technique to them. The rich picture includes all the important hard 'facts' of the organisational situation. However, these are not the only important facts. There are many soft or subjective 'facts' which should also be represented, and the process of creating the rich pictures serves to tease out the concerns of the people in the situation. These soft facts include the sorts of things that the people in the problem area are worried about, the social roles which the people within the situation think are important, and the sort of behaviour which is expected of people in these roles.

Typically, a rich picture is constructed first by putting the name of the organisation that is the concern of the analyst into a large 'bubble', perhaps at the centre of the page. Other symbols are sketched to represent the people and things that inter-relate within and outside that organisation. Arrows are included to show these relationships. Other important aspects of the human activity system can be incorporated. Crossed-swords indicate conflict and the 'think' bubbles indicate the worries of the major characters.

Figure 5 represents a rich picture for a professional association. If it has been well drawn, you should get a good idea of who and what is central to the organisation and what are the important relationships.

The act of drawing a rich picture is useful in itself because:

- Lack of space on the paper forces decisions on what is really important (and what are side issues or points of detail for further layers of rich pictures);
- It helps people to visualise and discuss their own role in the organisation;
- It can be used to define the aspects of the organisation which are intended to be covered by the information system; and
- It can be used to show up the worries of individuals and potential conflicts.

Differences of opinion can be exposed, and sometimes resolved, by pointing at the picture and trying to get it changed so that it more accurately reflects people's perceptions of the organisation and their role in it.

Lesson 4: the political dimension is important. The manipulation of power, that is, the political dimension, is important in real-world situations. This transcends the rationale of any methodology. Most of the cases showed decisions being made which were influenced by considerations beyond those that are implied by the Multiview methodology. For example, in one application the choice of equipment was strongly influenced by the desire in the organisation to 'buy British'. In another example, hardware had been purchased before the analysis was concluded. This may be undesirable, but it is difficult for analysts to argue to management that such

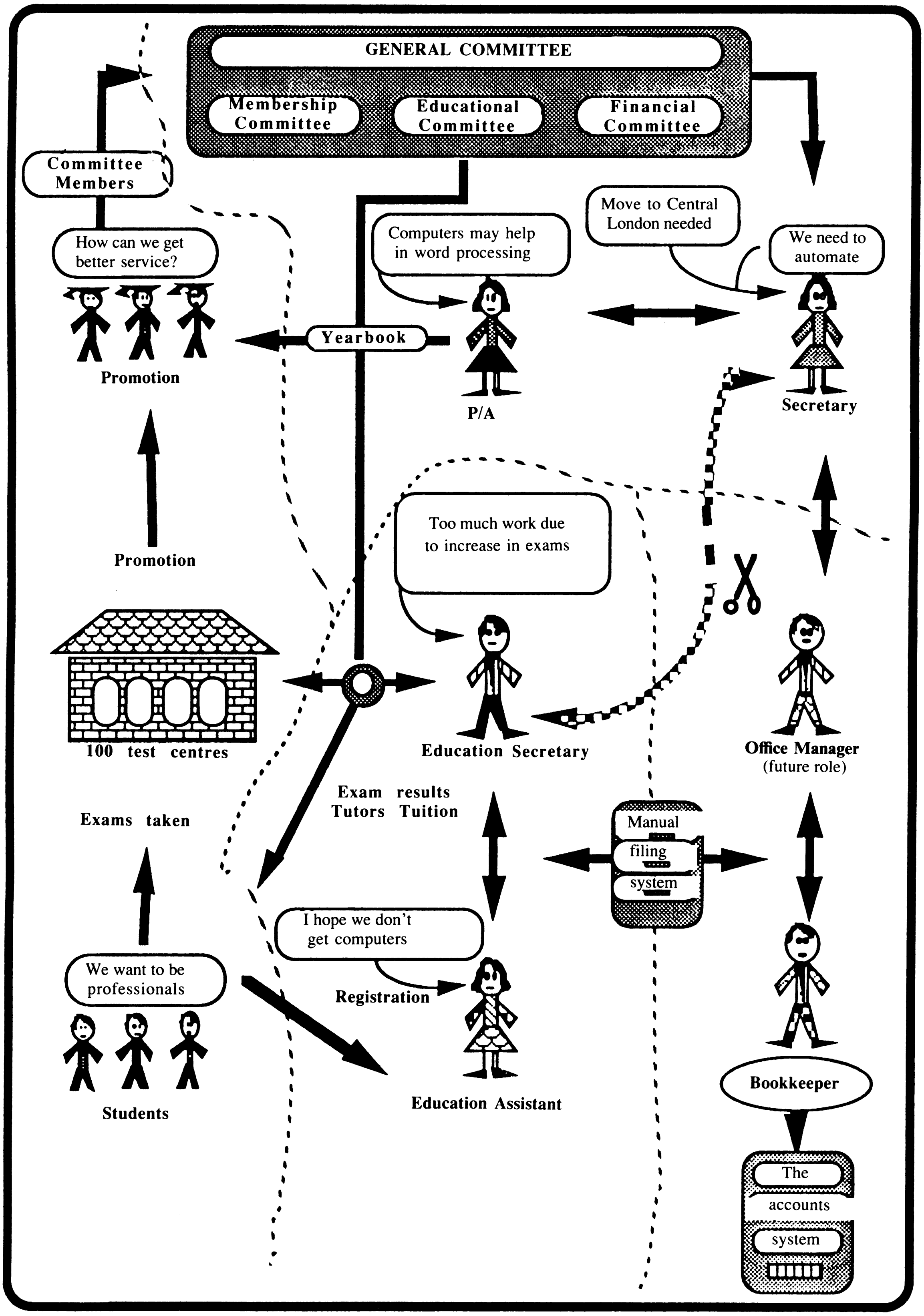

Fig. 5. Part of the rich picture relating to a professional association

purchases should not have been made – that they have made bad decisions, indeed, post-rationalisation is often expected from analysts.

Lesson 5: responsible participation is contingent. A high level of responsible participation, where appropriate, is a positive ingredient of successful information systems development. In the applications using Multiview, the role of the facilitator was frequently that of 'confidence booster' rather than that of adviser or applications developer. The role of the facilitators proved crucial in most applications and it is at least as much on the people side of systems than on the technical side.

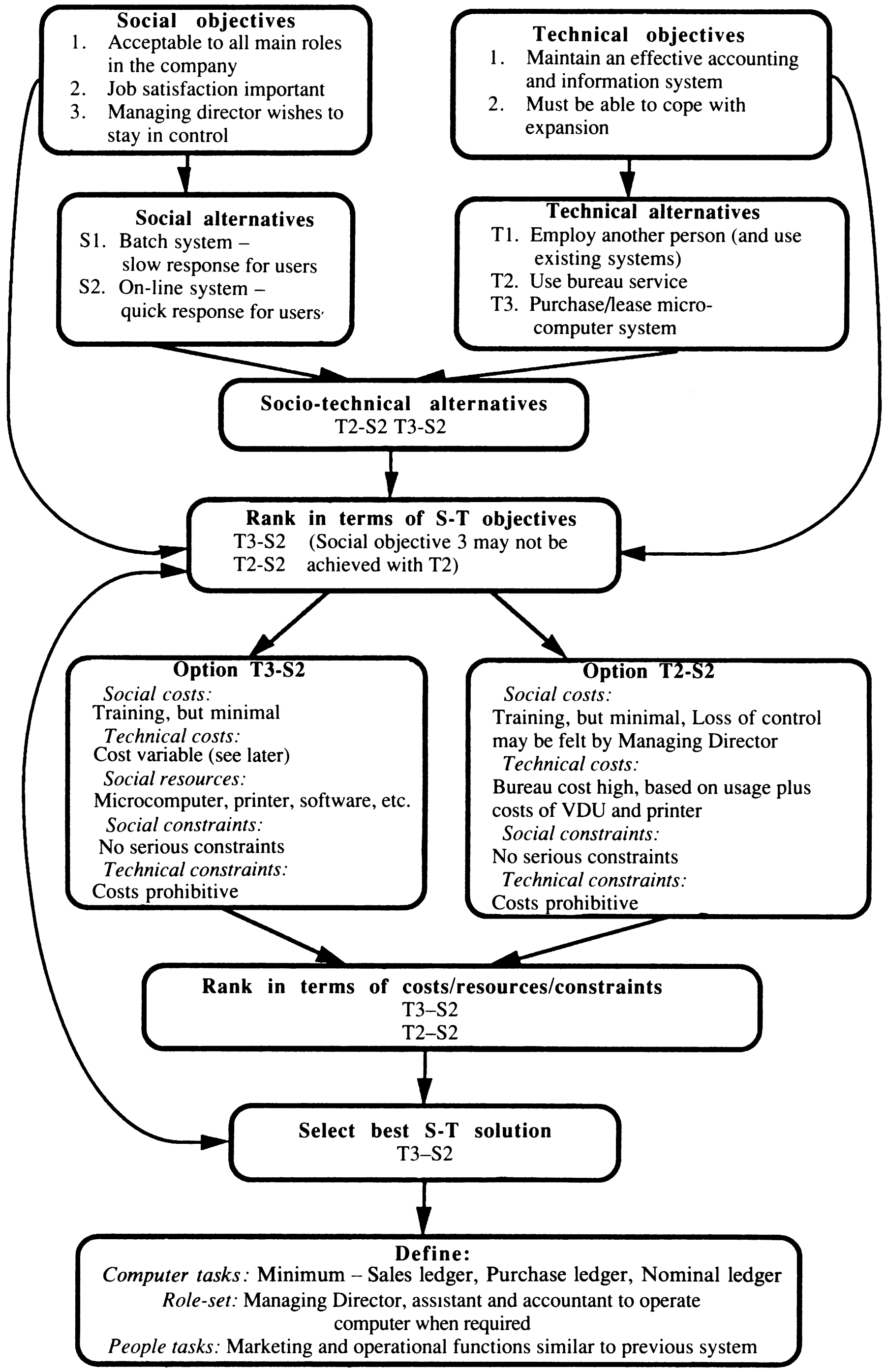

Fig. 6. Socio-technical outline for a freight agency

Sometimes participation is less successful. A key user in one application refused to cooperate fully. A possible interpretation of this behaviour was that the particular user wanted to develop his own system. The systems analysts developing an information system with other users therefore 'invaded' his territory. In other applications, managers have refused to cooperate, arguing that it was the job of systems analysts to develop the applications and they did not want to get involved. This contradicts the arguments of 'pure' Multiview, in which it is assumed that it is always possible to use responsible participation in information systems development.

Lesson 6: the technical dimension is also important. In the exposition of Multiview, the social and human dimension has been stressed. This is neglected in many information systems development methodologies. However, in Multiview we have attempted a balanced approach following, amongst others, the proponents of the socio-technical view.[66] Figure 6 shows a socio-technical outline for a freight agency where the various social and technical objectives and alternatives are listed separately and weighted, then combined, and finally possible socio-technical options ranked.

Although the social and human aspects of systems development are stressed in Multiview, sometimes the use of Multiview has exposed difficulties over technical decisions. Either the techniques were not well applied or there were major problems with the hardware and software used. For example, difficulties have been experienced interfacing modules, such as the database system, computer and optical mark reader in one application. According to the suppliers, this was a trivial matter, but software had to be written and the testing of the interface delayed the implementation of the prototype. In another application, there were major problems relating to the use and the limitations of the applications generator adopted. Programs were difficult to adapt to the specific needs of the application because the code generated by the package was difficult to follow.

In yet another application, many weeks were lost in the development of the prototype over the problems of using database management systems. One, for example, worked well until the number of files became very large, at which point it failed to work. Such limitations are only revealed after using the system 'in anger', that is with significant real data and processing.

Lesson 7: evaluation is difficult. In the first of these 'lessons', we argued that a blended contingency approach to information systems can work in practice and evidenced this by arguing that prototypes have been developed which were useful. This is a somewhat unsatisfactory way of justifying the approach, though evaluation is difficult in information systems work. Evaluation of an information system occurs most frequently at the early stage for cost-justification purposes and following implementation.[45] Evaluation in Multiview is also presently confined to a justification stage (albeit taking a much broader view of costs and benefits than that found in most methodologies) and post-implementation (again taking a broad view). However, it might be possible to include evaluation throughout the methodology and integrated in it. This would be desirable because that taking place at the feasibility study stage is prone to inaccuracy because it occurs so early in the life of a project (the design decisions have not been made and the information available is poor). At a post-implementation stage there may still be teething problems and unfamiliarity with the system or, if left later, problems which might have been resolved are left for some time during the operation of the system. It might therefore be desirable to have frequent monitoring of the system. One possible form that this could take is a benefits realisation programme.[7,78]

In one application we gave out questionnaires after implementing prototype systems. The questionnaire had four sections: effectiveness (objectives fulfilled, problems addressed, needs addressed, costs and achieved benefits, impact on work and clients, and impact on decision-making and control); efficiency of the system (attributes of the information produced, aspects of using the information system, value placed on information generated, data collection methods, working practices of staff, and the personal aspirations of staff); the design process (in terms of perceived 'success' of the implementation, contribution made by 'participation' procedures, and personal benefits and gains of staff); and finally the training process. On working practices, comments such as 'it was satisfying to see what work had been done at the end of the month', 'I have been made more aware of what I have been doing' and 'I have realised what a large amount of time is spent travelling and walking to homes and schools' are revealing.

Figure 7 is a Likert-type diagram looking at the various views of the users concerning the use of the new system. The crosses represent the average position of the user responses. This is revealing because it is critical of the new system. It highlights the time necessary to use the new system, though on follow-up interviews this was seen as a 'once-and-for-all' investment of time.

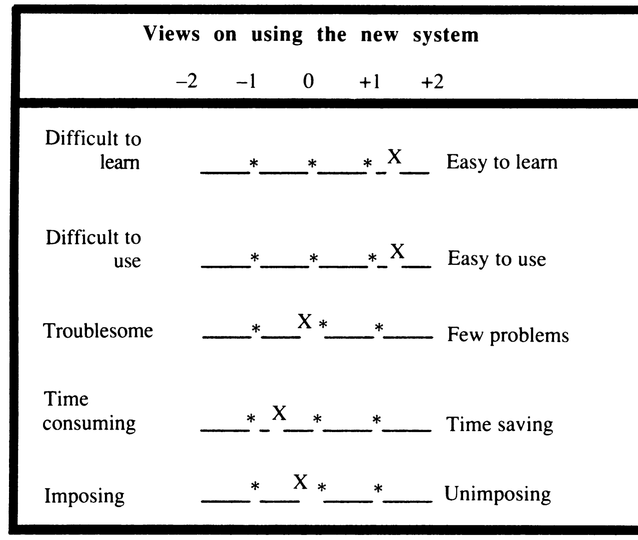

Fig. 7. User views for the new system

As for the method of implementation, one user commented: '... as it is us who have to use the system, we are in the best place to help design it; we know what information is important and relevant for recording...' and another: 'I have been made aware of the difficulties in the system design process' and another: 'I have improved my knowledge of computer systems, and found this experience both interesting and beneficial'.

There are dangers in any type of evaluation and any method should be cross-referenced using results from another method. User reactions described above may not represent the situation accurately.[84] The reaction of a football manager when the team wins or loses provides an illustration. These small events in a football season bring about extreme and opposite reactions when judging the team. Similarly, user reactions may be determined by the most recent experience with the system. Further, what may look good from the point of view of the user, may not be satisfactory from the analyst's view (or the problem owner). Time is also a major factor in evaluation. A follow-up discussion with users carried out in June 1990 at one organisation showed that the enthusiasm placed on the prototype in 1987 was no longer felt for the operational system!

Conclusion 1: the Multiview methodology is in a continuing state of development. This conclusion is not an attack on Multiview in isolation – all information systems development methodologies have limitations. Information Systems is a comparatively new discipline, the diversity of approaches is caused to some extent by the background and cultures of their authors and none is all-inclusive. The methodologies address a moving target in that the technology, along with techniques and tools supporting it, develop relentlessly. We have identified areas which do need to be addressed further, such as the problem of evaluating information systems and, perhaps, including evaluation as a permanent aspect of information systems development. Thus Multiview is part of a process of improving information systems practice. Further, because of the complexity of information systems development methodologies, the techniques and tools need to be explained in a way that enhances ease of learning and use. The difficulty in use may be partly because we have stressed the organisational and human aspects of information systems development more than the technical and technological aspects.

In a previous paper[3] it is argued that Multiview is an exploration in the development of information systems rather than a methodology, because the latter term implies a formal, fixed and inflexible approach. The writers of this paper have had differing experiences when using Multiview. Although there is always a 'family resemblance' in each version of the methodology (for example the basic structure of the 1985 and 1990 texts are the same), the 'agreed methodology in 1985' has been modified on the basis of these experiences, the literature and hardware and software developments since that time. Our experiences have been different, because no situation is the same as any other. The methodology will develop further in the future. New tools are likely to be incorporated, for example, though it is expected that the basic five phase framework, which provides the coherence in the approach, will be maintained.

Conclusion 2: developing an information system is contingent. Developing an information system is contingent on the methodology, the situation and the information systems development team. The team of users and analysts affect the perception of the situation and they interpret the methodology. The variety of possible interpretations reflect differences in the backgrounds and experiences of the analysts. The emphasis placed on various techniques, for example those user oriented and those computer oriented, will be biased by the background of the users. In some of the applications not all of the stages of Multiview were used because of the situation. It is possible to envisage cases where it is deemed inappropriate to develop a computer-based information system. The framework was adapted in one case. In other, the different analysts interpreted the 'same situation' differently. In any situation where an information system might be appropriate, there are factors such as culture, language and education which have to be taken into consideration. Sometimes the political and social climate is such that participation is difficult to achieve. In other situations particular techniques and tools are not appropriate to the problem situation. The systems analyst has to choose from a 'tool box' those techniques and tools appropriate for each situation, but within the framework of an approach such as Multiview.[6] Without such a framework, the information systems are likely to be idiosyncratic and difficult to maintain, and therefore of variable value. The decisions relating to choosing which tools, when and how to use them, is likely to be difficult without some basic structure.

Conclusion 3: the adoption of a contingency approach can lead to problems. Although developing an information system is contingent, there are a number of difficulties associated with adopting a contingency approach. These include factors relating to choice and factors relating to use:

- The first concerns the choice of approach to adopt (that is whether to follow the view of Davis[34] or that if Iivari[50] described earlier and then, respectively, which range of methodologies or which aspects of the framework to include.

  The complexity of information systems methodologies would seem to go against Davis's view that information systems departments ought to have available (and analysts be familiar with) a number of methodologies, any of which might be chosen for a given situation. It is too ambitious, perhaps, to expect systems analysts to know many methodologies, with different philosophies, as well as different techniques and tools, covering different parts of the life cycle.

  In Multiview we have constructed a methodology framework following Iivari (rather than proposing a number of methodologies to be used as appropriate following Davis). We have made choices over the techniques and tools to be included, and are aware of the need to re-evaluate these as new ones become available or are discussed in the literature. Although there are difficulties with the Iivari approach, the experience gained from the cases would suggest that Multiview was a useful implementation of a contingency approach to information systems development.
- The second concerns difficulties associated with the levels of expertise and the breadth of knowledge required from systems analysts in order to use these approaches. In a conventional information systems development methodology analysts are expected to follow a particular structure for all cases. This structure is well-defined and the techniques and tools within them are also well-defined.

  The analyst following an approach such as Multiview needs to know which technique and tool from a range might be appropriate for a particular situation at a particular point of time. This requires wide experience in the practice of systems analysis. The analyst also needs to know a broader range of techniques and tools from which to choose. In both cases, the analyst is helped by the Multiview framework which will guide the analyst in this choice. As stated earlier, this is preferred to a totally free choice. Nevertheless, considerably skill and experience is required to make these decisions.
- The third concerns difficulties relating to the consistency of standards in organisations that adopt a contingency approach. Because of the nature of contingency approaches, information systems will be developed using different techniques and tools and in different ways, dependent on the particular situation. This does mean a loss of common standards to some extent and this is one of the practical benefits that the use of an information systems development methodology is supposed to provide.

  This has been recognised by us and in the latest version of Multiview, documentation and other standards are recommended. Further, the techniques (such as data flow diagrams and entity life histories) are now more in line with the best practice of information systems. It is necessary to continue developing Multiview in order to keep it in line with information systems knowledge and to continue using Multiview in different problem situations.
- The fourth concerns the identification of the various elements of contingency and their importance. In

general terms, we have suggested that the information systems development process is dependent on the methodology, the problem situation and the team of analysts and users developing the system. Contingency factors relating to the team of analysts and users, for example, will include their training, education, culture, experience and so on. But issues relating to when and why these factors are important and how they influence the success of the information system have not been fully explored. Similar difficulties relate to how an analyst chooses from the 'tool box' in any particular problem situation, as it is not always clear under what circumstances a particular technique or tool is advised or ill-advised.

Conclusion 4: there are problems as well as advantages of action research in information systems. The authors have developed Multiview in the tradition of action research. Action research allows the researcher great potential to utilise the ideas of users and change concepts and methods as the work develops. Researchers and subjects cooperate in solving a real-life problem. It is particularly useful that action research allows work to take place in its natural setting and it gives the researcher an insight into real-life practical areas. Feedback from the practical application of techniques can be used to redefine and improve those techniques and their description. Although the results of action research are of a qualitative nature, they do offer a degree of external validity because the theory developed can be interpreted and refined by others in other real-world situations.

Action research has proved helpful to the development of Multiview. The weaknesses in the descriptions of some of the techniques, such as data flow diagrams and entity life histories, was revealed when using them in early applications. The practicality of using techniques and tools contingent on a particular problem situation as suggested by Multiview, can only be revealed by its use in different situations. Major omissions in the exposition of Multiview, such as the inadequacy of documentation procedures, project control and the lack of advice when selecting application packages, were revealed in the cases. Some assumptions in Multiview, such as the users' enthusiasm for participation and the low relative importance of the technical dimension have also been questioned through this experience. The part played by researchers as active players in the problem situation (not merely 'impartial' observers), along with users and analysts, has meant that both researchers and practitioners have together influenced the practice (by implementing change in the problem situation) and the theory (by changing the Multiview approach).

However there is presently a debate concerning information systems research,[12] and there are disadvantages of action research. The lack of impartiality of the researcher has led to its rejection by a number of researchers and academic departments. The lack of scientific discipline in such research makes it difficult for the work to be assessed for the award of research degrees and for publication in academic journals. A particular difficulty that universities have is persuading research funding bodies that this type of research is as valid and as useful as conventional methods of scientific research. Further, although the researcher's intent is to conduct research while effecting change, the approach is sometimes branded with the description 'consultancy' and not research. The open-endedness of such research and the consequent flexibility necessary in writing a research proposal also provide additional difficulties. Further, a major consequence of the choice of the action research method is that the research is context-bound as opposed to context-free. It is difficult to determine the cause of a particular effect, which could be due to environment (including its subjects), researcher, or methodology. This has been explicit in this research, but it can mean that action research produces narrow learning in its context because each situation is unique and cannot be repeated. This is a major criticism of action research – it does not produce generalisable learning.

In this research there is an attempt to reconcile the narrow learning from action research with the need for generalisable research.[91] It is hoped that the generalised findings from our work will result in other researchers and practitioners applying the Multiview theory in other problem situations and will take into consideration the learning that emerges in the complex process of developing an information system.

REFERENCES

1. L. Antill and A. T. Wood-Harper, The who's doing what for whom design stage. *SOFT* **2** (2) (1984).
2. D. E. Avison, *Information Systems Development: A Data Base Approach*. Blackwell Scientific Publications, Oxford (1985).
3. D. E. Avison and A. T. Wood-Harper, Multiview – An exploration in information systems development. *Australian Computer Journal* **18** (4) (1986).
4. D. E. Avison and G. Fitzgerald, *Information Systems Development: Methodologies, Techniques and Tools*. Blackwell Scientific Publications, Oxford (1988).
5. D. E. Avison and G. Fitzgerald, Information systems development: current themes and future directions. *Information and Software Technology* **30** (8) (1988).
6. D. E. Avison, G. Fitzgerald and A. T. Wood-Harper, Information systems development: a tool-kit is not enough. *Computer Journal* **31** (4) (1988).
7. D. E. Avison, C. P. Catchpole and J. Horton, *Achieving Benefits from Health Information Systems*. MEDINFO89, Sixth World Conference on Medical Informatics, Singapore (1989).
8. D. E. Avison, Action learning for information systems teaching. *International Journal of Information Management* **9** (1) (1989).
9. D. E. Avison and A. T. Wood-Harper, *Multiview: An Exploration in Information Systems Development*. Blackwell Scientific Publications, Oxford (1990).
10. D. E. Avison, A departmental information system. *International Journal of Information Management* **10** (2) (1990).
11. D. E. Avison, A Contingency Framework for Information Systems Development. PhD Thesis. Aston University, Birmingham (1990).
12. D. E. Avison and G. Fitzgerald, Information systems practice, education and research. *Journal of Information Systems* **1** (1) (1991).
13. D. E. Avison and D. Wilson, Controls for effective prototyping. *Journal of Management Systems* **3** (1) (1991).

14. S. Beer, *Diagnosing the System for Organizations*. Wiley, Maidenhead (1985).
15. T. M. A. Bemelmans, *Beyond Productivity: Information Systems Development for Organizational Effectiveness*. North Holland, Amsterdam.
16. D. Benyon and S. Skidmore, Towards a tool-kit for the systems analyst. *Computer Journal* **30** (1) (1987).
17. L. von Bertalanffy, *General Systems Theory*. Braziller, New York (1968).
18. N. Bjørn-Andersen, *Challenge to Certainty*. In Bemelmans (1984).
19. B. H. Boar, *Application Prototyping: A Requirements Definition Strategy for the 80's*. Wiley, New York (1984).
20. J. A. Bubenko Jr. *Information System Methodologies – A Research View*. In: Olle et al (1986).
21. R. Budde, K. Kuhlenkamp, L. Mathiassen and H. Zullighoven, *Approaches to Prototyping*. Springer-Verlag, Berlin (1984).
22. R. N. Burns and A. R. Dennis, Selecting the appropriate application development methodology. *Data Base*, **17** (1) (1985).
23. D. C. Burnstine, *BIAIT: An Emerging Management Engineering Discipline*. BIAIT International, Petersburg, New York (1986).
24. L. Capper, *The Use of Analysis and Design Methodologies in the Working Environment: An Experimental Approach*. In Bemelmans (1985).
25. W. M. Carlson, Business Information Analysis and Integration Technique (BIAIT), The New Horizon. *Data Base*, **10** (4) (1979).
26. P. B. Checkland, *Systems Thinking, Systems Practice*. John Wiley, Chichester (1981).
27. P. B. Checkland and J. Scholes, *Soft Systems Methodology in Action*. John Wiley, Chichester (1990).
28. P. B. Checkland, *Rethinking a Systems Approach*. In Tomlinson & Kiss (1984).
29. J. D. Cougar and R. W. Knapp, *Systems Analysis Techniques*. Wiley, New York (1974).
30. J. D. Cougar, M. A. Colter and R. W. Knapp, *Advanced Systems Development/Feasibility Techniques*. Wiley, New York (1982).
31. G. Curtis, *Business Information Systems: Analysis, Design and Practice*. Addison-Wesley, Wokingham (1989).
32. A. Daniels and D. A. Yeates, *Basic Training in Systems Analysis*, 2nd ed. Pitman, London (1971).
33. L. J. Davies and A. T. Wood-Harper, Information systems development: Theoretical frameworks. *Journal of Applied Systems Analysis* **16** (1989).
34. G. B. Davis, Strategies for information requirements determination. *IBM Systems Journal* **21** (2) 4–30 (1982).
35. G. B. Davis and M. H. Olsen, *Management Information Systems: Conceptual Foundations, Structure and Development*, 2nd ed. McGraw-Hill, New York (1985).
36. T. DeMarco, *Structured Analysis and System Specification* Prentice Hall, Englewood Cliffs (1979).
37. E. Downs, P. Clare and I. Coe, *Structured Systems Analysis and Design Method: Application and Context*. Prentice Hall, Hemel Hempstead (1988).
38. D. M. Episkopou and A. T. Wood-Harper, Towards a framework to choose appropriate IS approaches. *Computer Journal* **29** (3) (1986).
39. R. Espejo and R. Harnden, *The Viable Systems Model*. Wiley, New York (1985).
40. G. Fitzgerald, N. Stokes and J. R. G. Wood, Feature analysis of contemporary information systems methodologies *Computer Journal* **28** (3) (1985).
41. L. M. Fok, K. Kumar and A. T. Wood-Harper, *Methodologies for Socio-Technical Systems (STS) Development: A Comparative Review*. 8th International Conference of Information Systems, Pittsberg, December (1987).
42. C. P. Gane and T. Sarson, *Structured Systems Analysis: Tools and Techniques*. Prentice Hall, Englewood Cliffs (1979).
43. C. B. B. Grindley, *SYSTEMATICS – A Nonprogramming Language for Designing and Specifying Commercial Systems for Computers*. In Cougar & Knapp (1974).
44. C. B. B. Grindley, *The Use of Decision Tables within Systematics*. In Cougar & Knapp (1974).
45. J. Hawgood and F. F. Land, *A Multivalent Approach to Information Systems Assessment*. In N Bjørn-Andersen and G. B. Davis, Proceedings of IFIP WG8.2 Conference on Information Systems Assessment, Noordwijkerhout in August 1986, North Holland, Amsterdam (1988).
46. S. Hekmatpour and D. Ince, *Rapid Software Prototyping*. Open University, Technical Report, 86/4 (1986).
47. R. Hirschheim and H. K. Klein, Four paradigms for information systems development. *Communications of the ACM*, **32** (10) (1989).
48. IBM (1971) *The Time Automated Grid System* (*TAG*). In Cougar & Knapp (1974).
49. IBM (1975) *Business Systems Planning*. In Cougar, Colter & Knapp (1982).
50. J. Iivari, *A Methodology for IS Development as an Organisational Change: a Pragmatic Contingency Approach*. In Klein & Kumar (1989).
51. H. Klein and K. Kumar, *Information Systems Development for Human Progress in Organisations*. North Holland, Amsterdam (1989).
52. F. F. Land and R. Hirschheim, Participative systems Design: rationale, tools and techniques. *Journal of Applied Systems Analysis* **10** (1983).
53. F. F. Land and E. Somogyi, Software engineering: the relationship between a formal system and its environment. *Journal of Information Technology* **1** (1) (1986).
54. F. F. Land, Book review of Curtis (1989). *Times Higher Education Supplement*, 29/9/90 (1990).
55. K. E. Lantz, *The Prototyping Methodology*. Prentice Hall, Englewood Cliffs (1985).
56. D. T. Law, *Prototyping*. NCC, Manchester (1985).
57. G. Longworth, *Designing Systems for Change*. NCC, Manchester (1985).
58. M. Lundeberg, G. Goldkuhl and A. Nilsson, *Information Systems Development – A Systematic Approach*. Prentice Hall, Englewood Cliffs, New Jersey (1982).
59. M. Lundeberg, *The ISAC Approach to Specification in Information Systems and its Application to the Organization of an IFIP Working Conference* In Olle *et al* (1982).
60. R. N. Maddison, *Information System Methodologies*. Wiley Heyden, Chichester (1983).
61. L. M. Markus, *Information Systems in Organisations: Bugs and Features*. Pitman, Marshfield (1984).
62. J. Martin, *Strategic Data Planning Methodologies*. Savant, Carnforth, Lancashire (1980).
63. J. Martin and C. Finkelstein, *Information Engineering*. Vols 1 and 2, Prentice Hall, Englewood Cliffs, New Jersey (1981).
64. P. J. Mayhew and P. A. Dearnley, An alternative prototyping classification. *Computer Journal* **30** (6) (1987).
65. E. Mumford, F. F. Land and J. Hawgood, A Participative approach to computer systems. *Impact of Science on Society* **28** (3) (1978).
66. E. Mumford and M. Weir, *Computer Systems in Work Design – The ETHICS Method*. Associated Business Press, London (1979).
67. E. Mumford, Participative systems design: Structure and method. *Systems, Objectives and Solutions*, 1. (1981).
68. E. Mumford, R. A. Hirschheim, G. Fitzgerald and A. T. Wood-Harper, (Eds) *Research Methods in Information Systems*. North-Holland, Amsterdam (1985).
69. G. J. Myers, *Reliable Software through Composite Design*. Petrocelli/Charter, New York (1978).
70. G. J. Myers, *Composite/Structured Design*. Van Nostrand Rinehold, New York (1978).

71. J. D. Naumann and A. M. Jenkins, Prototyping: The new paradigm for systems development. *MIS Quarterly* **6** (3) (1982).
72. NCC *SSADM Manual*. National Computing Centre, Manchester (1986).
73. G. M. Nijssen and S. M. Twine, *A Framework for Information Systems Development Methodologies*. Prentice-Hall, Sydney (1991).
74. T. W. Olle, H. G. Sol and A. A. Verrijn-Stuart, *Information Systems Design Methodologies: A Comparative Review*. North Holland, Amsterdam (1982).
75. T. W. Olle, H. G. Sol and C. J. Tully (Eds), *Information Systems Design Methodologies: A Feature Analysis*. North Holland, Amsterdam (1983).
76. T. W. Olle, H. G. Sol and A. A. Verrijn-Stuart (Eds). *Information Systems Design Methodologies: Improving the Practice*. North Holland, Amsterdam (1986).
77. T. W. Olle, J. Hagelstein, I. G. Macdonald, C. Rolland, H. G. Sol, F. J. M. Van Assche and A. A. Verrijn-Stuart, *Information Systems Methodologies: A Framework for Understanding*. Addison-Wesley, Wokingham (1988).
78. R. Parker, Realizing benefits: The role of management and the computer. *Computers and Healthcare* **9** (3) (1988).
79. P. T. Quang and C. Chartier-Kastler, *Merise in Practice*. Macmillan, Basingstoke (translated by D. E. and M. A. Avison from the French *Merise Appliquée*. Eyrolles, Paris, 1989). (1991).
80. H. Robinson, *Database Analysis and Design*, 2nd ed. Chartwell-Bratt, Bromley (1989).
81. R. Rock-Evans, *Data Analysis*. IPC Press, London (1981).
82. R. Rock-Evans, *CASE Analyst Workbenches: A Detailed Product Evaluation*. Ovum Report, London (1989).
83. M. J. R. Shave, Entities, functions and binary relations: steps to a conceptual schema. *Computer Journal* **24** (1) (1989).
84. D. Silverman, Six rules of qualitative research: A post-romantic argument. *Symbolic Interaction* **12** (2) (1989).
85. D. Teichroew and E. A. Hershey, PSL/PSA: A Computer-Aided Technique for Structured Documentation and Analysis of Information Processing Systems. In Cougar *et al* (1982). (1977).
86. D. Teichroew, E. A. Hershey and Y. Yamamoto, *The PSL/PSA Approach to Computer-Aided Analysis and Documentation*. In: Cougar *et al* (1982). (1979).
87. R. Tomlinson and I. Kiss, *Rethinking the Process of Operational Research and Systems Analysis*. Pergamon, Oxford (1984).
88. G. Vickers, *Freedom in a Rocking Boat*. Allen Lane, London (1970).
89. G. Vidgen and J. Hepworth, Yesterday's philosophy. *British Journal of Healthcare Computing* **7** (7) (1990).
90. A. A. Verrijn-Stuart, Themes and trends in information systems TC8: 1975–1985. *Computer Journal* **30** (2) (1987).
91. A. Warmington, Action research: Its methods and its implications. *Journal of Applied Systems Analysis*, 7. (1980).
92. V. Weinberg, *Structured Analysis*. Prentice Hall, New Jersey (1978).
93. R. J. Welke, *The New Architecture of PSL/PSA*. Ann Arbor, MI: MetaSystems (1978).
94. J. C. Wetherbe and G. B. Davis, *Developing a long-range information architecture*. Proceedings of the National Computer Conference, Anaheim, Calif, May 1983 (1983).
95. B. Wilson, *Systems: Concepts, Methodologies and Applications*. 2nd ed. Wiley, Chichester (1990).
96. A. T. Wood-Harper and G. Fitzgerald, A taxonomy of current approaches to systems analysis. *Computer Journal* **25** (1) (1982).
97. A. T. Wood-Harper and D. J. Flynn, Action learning for teaching information systems. *Computer Journal* **26** (1) (1983).
98. A. T. Wood-Harper, *Information Research Methods: Using Action Research*. Mumford *et al* (1985).
99. A. T. Wood-Harper, L. Antill and D. E. Avison, *Information Systems Definition: the Multiview Approach*. Blackwell Scientific Publications, Oxford (1985).
100. A. T. Wood-Harper and S. W. Corder, *Seeds of Ideas in Information Systems*. In Klein & Kumar (1989).
101. A. T. Wood-Harper, Comparison of information systems definition methodologies: action research multiview perspective. PhD Thesis, University of East Anglia, Norwich (1989).
102. S. B. Yadav, Determining an organisations information requirements: a state of the art survey. *Data Base* **14** (3) (1983).
103. E. Yourdon, *Modern Structured Analysis*. Prentice Hall, Englewood Cliffs (1989).
104. E. Yourdon and L. L. Constantine, *Structured Design*, 2nd. ed. Yourdon, New York (1978).

Wilkes Award

The Society is pleased to announce that the Wilkes Award for 1989 has been won by Stuart C. Wray and Jon Fairbairn for their paper 'Non-strict Languages – Programming and Implementation', which was published in the April 1989 issue (vol. **32**, no. 2) of *The Computer Journal*. Each year an Award Panel considers all those papers, published in the previous year, where one or more of the authors was under thirty years of age at the time of submission. The award is then made to the author(s), provided they satisfy the age criterion, of the best paper in that category.

The paper addresses a very important problem of functional programming, namely the contradiction between creating efficient implementations of functional languages and producing elegant programs – which still need to be implemented efficiently but currently are not. The paper explains the problem and goes on to describe a method for implementing functional languages which allows elegant programs to be implemented efficiently. The paper was chosen to receive the award for two reasons. The first was the clear way in which the problem was described and in which the solution was both achieved and described. The second was that the panel of judges felt that the work described in the paper would have more impact on the future directions of computer programming than that of the other papers under consideration.

Stuart Wray was born in Knaresborough and went up to Cambridge in 1978 to study physics. At the end of the first year he transferred to the Computer Science Tripos. After graduating he spent one year working with Torch Computers in Cambridge and then returned to the University to complete a PhD thesis on the Efficient Implementation of Programming Languages. He then spent one year as an SERC research fellow before joining Olivetti Research in 1987. At Olivetti he is currently working on object-based mechanisms for controlling networked digital video. Within this work he has retained his interest in using functional languages efficiently.

Jon Fairbairn went up to Cambridge from Scarborough Sixth Form College, in between spending one year with Marconi Research and Marconi Radar Systems Limited. He took his PhD in 1985 and then, like Stuart Wray, became an SERC research fellow in the University. There followed a number of visits to Glasgow University, Chalmers University of Technology and the University of Calgary. He returned to the University of Cambridge in 1989 with a Royal Society University Research Fellowship. This has enabled him to continue his research on types.

Jon Fairbain and Stuart Wray were two of 46 authors who were considered for the award. The Wilkes Award, which consists of a silver gilt medallion, was instituted by the British Computer Society to mark the retirement of Professor M. V. Wilkes as Professor of Computing Technology of the University of Cambridge, in recognition of his pioneering work in both computer hardware and software and his unstinting efforts on behalf of the Society.