Task allocation and job specification in computer systems design

This item was submitted to Loughborough University's Institutional Repository by the/an author.

Additional Information:

- A master's thesis submitted in partial fulfilment of the requirements for the award of Master of Philosophy at Loughborough University.

Metadata Record: [https://dspace.lboro.ac.uk/2134/31847](https://dspace.lboro.ac.uk/2134/31847)

Publisher: © Kelvin Ho

Rights: This work is made available according to the conditions of the Creative Commons Attribution-NonCommercial-NoDerivatives 2.5 Generic (CC BY-NC-ND 2.5) licence. Full details of this licence are available at: [http://creativecommons.org/licenses/by-nc-nd/2.5/](http://creativecommons.org/licenses/by-nc-nd/2.5/)

Please cite the published version.
This item was submitted to Loughborough University as an MPhil thesis by the author and is made available in the Institutional Repository (https://dspace.lboro.ac.uk/) under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
<table>
<thead>
<tr>
<th>AUTHOR/FILING TITLE</th>
<th>Ho K K W</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCESSION/COPY NO.</td>
<td>0360000934</td>
</tr>
<tr>
<td>VOL. NO.</td>
<td>CLASS MARK</td>
</tr>
<tr>
<td>Loan copy</td>
<td></td>
</tr>
<tr>
<td>10 MAR 1995</td>
<td></td>
</tr>
<tr>
<td>28 JUN 1996</td>
<td></td>
</tr>
<tr>
<td>22 MAR 1996</td>
<td></td>
</tr>
<tr>
<td>13 JUN 1997</td>
<td></td>
</tr>
</tbody>
</table>
Task Allocation and Job Specification

in Computer Systems Design

by

Mr. Kelvin Ka Wai Ho  B.Sc.(Hons)

A Master's Thesis
Submitted in partial fulfilment of the requirements
for the award of

Master of Philosophy

of

the Loughborough University of Technology

April 1991

© by Kelvin Ho 1991
Abstract

There are many problems to resolve during the development of any computer system. Different methodologies have been devised and developed to deal with these problems. Many of the commonly used structured design methodologies and their methods are mainly data-driven and technically oriented. They fail to address the issue of task allocation and job design requirements for the end users. They do not see the whole organization as an open and complex socio-technical system, with human activity as an important part of it. This often results in systems which meet the functional specifications, but fail to satisfy the social or human requirements as well as staff aspirations. This generates dissatisfaction amongst the users, thus reducing the usability and overall effectiveness of the system. There have been many attempts to incorporate job design into system analysis. One of them is the DIADEM methodology, in which the technique of Job Stream Charting was developed to include considerations of job design issues early in the design process. The DIADEM methodology has been used in a very large structured organization where a new computer system was required. This project investigates the application of the Job Stream Charting technique in the DIADEM project and the problems encountered. The project identifies a lack of coordination and communication between the social and technical design teams regarding their respective work. There were also difficulties in interpreting and making use of the charts from Job Stream Charting. This project suggests improvements and demonstrates their feasibility by examples.
Acknowledgements

I would like to thank my supervisor, Dr. Steve Scrivener, for his guidance, patience and wisdom throughout the research programme. I would also like to thank Professor Ken Eason and Mrs. Susan Harker of the Department of Human Sciences for their invaluable advice, and to Mrs. Wendy Olphert of the Human Sciences and Advance Technology (HUSAT) Research Institute for her assistance. A special thanks to Dr. Helmut Bez, Mrs. Judith Poulton, and the Interactive Computing Systems Design course students for providing me with valuable information.

There are of course many others who have helped me during the research programme in many different ways. If I were to mention all of them, it would have run into another page and I would still be in danger of omitting someone. Therefore, all I can do is to express my deepest gratitude to all of you.

Finally, I would like to thank my family for their continuous support and encouragement, without which I could never have completed the research.

This thesis is dedicated to my grandfather, who passed away before witnessing its completion. I hope that somewhere, somehow, he knows that his grandson has at last completed his academic endeavors.
Abstract
Acknowledgements
Content
Introduction .................................................. 1
1. System Analysis ........................................... 5
2. Information System Methodologies ....................... 10
   2.1 The Need for Methodologies .......................... 10
   2.2 Technically Oriented Methodologies .................. 12
      2.2.1 Conventional System Analysis .................... 12
      2.2.2 Structured Systems Analysis ...................... 13
         2.2.2.1 SSADM .................................. 16
         2.2.2.2 The Use of SSADM in the UK ................ 19
      2.2.3 Limitations in Structured Analysis Methods .... 21
      2.2.4 Summary .................................... 25
   2.3 Socially Oriented Methodologies ..................... 26
      2.3.1 Participative Design ................................
         2.3.1.1 ETHICS ................................ 28
      2.3.2 Experience With Participative Design ........... 29
   2.4 Other Methodologies .................................. 33
      2.4.1 Soft Systems Methodology ....................... 33
      2.4.2 Prototyping .................................. 35
   2.5 Combination Methodologies ............................ 38
      2.5.1 DIADEM ..................................... 39
      2.5.2 User-Centred Method ............................ 41
      2.5.3 Multiview .................................... 44
   2.6 Summary ............................................ 47
3. Methodology Comparison ................................... 50
   3.1 Problems with Comparison ............................. 50
   3.2 Methods of Comparison ................................ 53
      3.2.1 Feature Analysis ................................ 53
         3.2.1.1 Maddison ................................ 53
         3.2.1.2 Catchpole ................................ 54
      3.2.2 Alternatives to Feature Analysis ................. 55
         3.2.2.1 Bubenko ................................ 55
         3.2.2.2 Davis ................................... 56
         3.2.2.3 Episkopou and Wood-Harper .................. 56
         3.2.2.4 NIMSAD .................................. 57
         3.2.2.5 Avison & Fitzgerald ......................... 58
      3.2.3 Summary .................................... 61
   3.3 A Comparative Study on Methodology .................. 61
   3.4 Social and Human Factors Techniques Contribution .. 64
   3.5 Experience in Integrating Human Factors Principles .. 66
   3.6 Summary ............................................ 69
4. Task Allocation and Job Specification

4.1 Introduction

4.2 Job Stream Charting

4.2.1 How to Construct the Charts

4.3 JSC Usage in DIADEM

4.4 JSC and the Design Teams

4.5 Application of JSC

4.5.1 Task Allocation

4.5.2 Job Stream Charts

4.5.3 Interface Design

4.6 Guidelines for JSC

4.7 Problems with JSC Application

4.7.1 General

4.7.2 Using JSC

4.8 Some Suggested Improvements

4.8.1 Inclusion of TACs

4.8.2 Cross Referencing

4.8.3 Task Catalog

4.8.4 Training

4.8.5 An Example

4.9 Summary

5. Incorporating Human Factors Techniques into Methodology

5.1 Management's Recognition

5.2 Improve Output Quality

5.3 More Human Scientist Involvement

5.4 Summary

Appendix A SSADM in Detail

Appendix B ETHICS in Detail

Appendix C DIADEM in Detail

Bibliography
Introduction

System analysis is an essential part in designing and building information systems, and system design methodologies have been developed over many years to assist analysts in carrying out a proper analysis and design. In this thesis we argue that there have been two basic trends in the development of analysis and design methodology. The first major trend is the development of technically oriented methodologies. The earliest of these methodologies was that documented in 1971 by the National Computing Centre (Avison 1985). However, the inflexibility of the method and its obvious failure to produce satisfactory solutions (Avison 1985) led to the development of methodologies aimed at structuring the analysis and design process. This development led to structured system analysis (Yourdon 1979, Gane and Sarson 1979), and on to one of the currently favoured methodologies, SSADM (Downs et al 1988, Ashworth and Goodland 1990). Nevertheless, these methodologies are concerned with improving the technical solution. The thesis argues, based on a report by Edwards et al (1989) on the experience of using SSADM by industry and another report by Vitalari (1984), that there are limitations in using structured analysis methodologies. These methodologies lack adequate provisions for the social and human factors aspects of system design. This led to a different trend of methodology.

This trend is the development of socially oriented methodologies. These methodologies allow the end users to be heavily involved in the design process from the beginning. An example of a socially oriented methodology is the participative design method, ETHICS (Humford 1986). Recently, there have been other alternative development of methodologies, such as Checkland's Soft Systems Methodologies.
(Checkland 1981) and prototyping (Harker 1988). There have also been developments that aim at combining and integrating aspects of the technical and social methodologies. Examples of such combined methodologies include DIADEM (Damodaran et al 1988, DIADEM 1987, Ip et al 1990), the User-Centred method (Eason 1988, Eason and Harker 1984) and the Multiview approach (Wood-Harper et al 1984).

Many of the methodologies discussed are theoretical methods of analysis. It is important therefore to see how the methodologies work in practice and how they compare with one another. It turns out that most methodologies used in practice are technically oriented. This may be due to the fact that these methods tend to be instructional and prescriptive. Other methodologies require consideration of social or human requirements, as well as an understanding of the philosophy of approach. The key point that emerges from a consideration of these developments is that most of the technical methodologies currently in use make little or no provisions for human factors contributions in system design. We will argue that methodological concerns are shifting towards socio-technical methods which integrate human factors and technical requirements together into a more complete design process. However, much of the work here is theoretical and it is important to investigate how such socio-technical methods work in practice. We will observe one such methodology, DIADEM, which integrates human factors principles into a structured design methodology, SSADM (Damodaran et al 1988, Ip et al 1990). We will highlight the problems encountered when using DIADEM through one of its techniques called Job Stream Charting (JSC): the inability to make use of JSC information together with other technical analysis, and the deficiencies in JSC as a representation. We will also suggest some improvements to the technique with illustrations.

2
In Chapter One, some of the problems faced by analysts are discussed, and the advantages of systematic analysis and design methodologies are highlighted. In Chapter Two, the need for a methodology is discussed, followed by a look at the different trend in the development of these methodologies. One way to see how effective these methodologies are in practice is to compare them. Chapter Three discusses the problems with comparing methodologies, followed by a discussion on the methods of comparison, highlighting several ways of carrying out comparison. After discussing how to compare methodologies, the next logical step is to look at actual comparisons that have been carried out. The Chapter highlights a comparative study carried out by the Berlin University on a group of technically based methodologies, and the conclusions from it. It is important to consider the effectiveness of socio-technical methodologies in practice to establish their usability and usefulness. In Chapter Four, the DIADEM methodology is taken as example of a socio-technical method used in a large UK organization. In particular one of DIADEM's techniques called Job Stream Charting is highlighted. This technique is specifically designed and developed to deal with job design and work organization issues. The thesis identifies the problems encountered during DIADEM's use and proposes and demonstrates solutions to these problems. In Chapter Five, the suggested improvements to Job Stream Charting are discussed at a more general level to see how human factors issues, such as job design can be incorporated into methodologies.
1. System Analysis

System analysis is an essential process in the construction of an information system. An information system is well-designed if it fulfills the following points (Connor 1985):

1) It satisfies the user's real information requirements,
2) It is easy to use and to operate,
3) It is easy to maintain (i.e. to correct problems that occur),
4) It is easily modified or enhanced.

However, analysts have to overcome many problems in order to produce such systems. Gane and Sarson (1979) highlight five aspects that are commonly encountered.

The first is to obtain the right system requirements from the users. The analyst has to have a clear explanation of system requirements from users, and cannot know what he has not been told. But the users do not necessarily know how to explain the way jobs are done and the way decisions are made in the present organization.

The second problem is that the users do not have sufficient knowledge, about the availability of technology, to know what is feasible or achievable and what is not. This may lead to the users requesting a system which cannot be designed and implemented because of hardware and software limitation or financial constraints.

The third problem is the collecting, organising and analyzing of the vast amount of information required by the analysts. This is the responsibility of the analysts and
their design teams, and must be carefully organised to avoid wasting time and resource even before problem solving starts.

The fourth problem is the apparent importance attached to the final system specification, or functional specification. This is often viewed as the "contract" that binds the users and the analyst team. Yet the users often do not understand the content of the document, or the implication of the system for their business once it is built and implemented. If any fundamental problems occur after implementation it will be too late to make radical changes.

The final problem is also to do with the functional specification of the system. Like the users, the physical designers and programmers may also not be able to use the specification for their purposes. Essentially, the physical design of the system should be left to those who have up-to-date technical knowledge, and should be based on an understanding of the complete logical requirements of the system provided by the analyst teams.

In order to tackle these problems, system analysis and design is generally carried out. However, Ashworth and Goodland (1990) point out that it is not easy to define what is gained from such analysis and design activities. However, by comparing the role of the system analyst to that of an architect, they come up with the following advantages. Carrying out analysis and design will:

1) ensure the involvement of end users at all stages of analysis and design activities to make sure that the design and specification of the system match user requirements, and a usable system is produced.
2) involve quality assurance whereby end products of each stage are examined by the analysts and users for quality, completeness, consistency and applicability. The approved products can then be passed on to developers and experienced systems staff to carry out the next stage.

3) separate the logical and physical specifications of the system, so that the best solution to the requirements is found first before entering into any physical design work. By producing the logical design, developers can tackle one problem at a time and prevent constraints appearing too early in development in the form of hardware or software dependency.

4) provide the opportunity to properly investigate what is required by the users. As a lot of assistance and encouragement is usually required to help the end users to describe their requirements, analysts can make use of their experiences and knowledge to obtain the requirements.

To solve these problems, methodologies are devised to help professional analysts and design teams to do their jobs as effectively and efficiently as possible. A methodology is a collection of procedures, techniques, tools and documentation aids that are helpful when developing a system (Avison & Fitzgerald 1988). Typically, it consists of phases and sub-phases which guide the developers in their choice of appropriate techniques. It also helps them to plan, manage, control and evaluate their projects. At this point, it is worth pointing out the distinction between a method and a methodology. A method will contain the features mentioned above and its use will be similar to using a recipe. A methodology will also contain these features, but it is also based on a view, a philosophy, of how systems should be
developed. Some may emphasize social and human aspects, others the technical requirements, others pragmatism, and yet others automated approaches. These different views will be characterized by the different assumptions that underpin them. These in turn will determine the techniques used in the methodology. Different methodologies will also have different objectives. Avison & Fitzgerald (1988) summarise some of them as follows:-

- To record accurately the requirements of an information system.
- To provide a systematic method of development in such a way that progress can be effectively monitored.
- To provide an information system within an appropriate time limit and at an acceptable cost.
- To produce a system which is well documented and easy to maintain.
- To provide an indication of any changes which need to be made as early as possible in the development process.
- To provide a system which is liked by those people affected by that system.

A definition of an information system methodology was suggested by Maddison in 1983 (Avison & Fitzgerald·1988). It is a recommended collection of philosophies, phases, procedures, rules, techniques, tools, documentation, management procedures and training programmes for developers of information systems. Avison & Fitzgerald also add that such a methodology should have a number of components which specify the following:-

- How a project is to be broken down into stages.
- What tasks are to be carried out at each stage.
- What outputs are to be produced.
- When the stages are to be carried out.
- What constraints are to be applied.
- What support may be utilized.
- How the project is to be managed and controlled.
- What type of support and training is needed for the users.

This chapter defines some of the problems encountered in system analysis, and some of the reasons for carrying out analysis and design. It also tries to make clear what an information system design methodology is. The next chapter will deal with the different theories underlying the methodologies as well as discussing the merits of some of the methodologies currently developed and, or, in use.
Current information system methodologies tend to concentrate on certain areas of system analysis and design. Wood-Harper et al (1985) suggest that there are three areas: those which put the emphasis on the functional or technical aspects of system design; those based primarily on the organizational or social changes and demands of the system; and those which combine other methods to cover all areas of system design. This section starts by discussing the need for these methodologies. It then sets out to describe several of the methods derived from the methodologies, and gives an outline of the aims and development of each method and how it works. It will also discuss some of the advantages and disadvantages of each method, based on available evidence and documented usage.

2.1 The Need for Methodologies

There are many methodologies currently used by many different organizations. Therefore it is useful to understand what the users are looking for in these methodologies. Avison and Fitzgerald (1988) suggest there are three main categories: a better end product, a better development process, and a standardized process.

Although every organization which embarks on a system development programme will want a better system to come out of the exercise, each will cherish different concepts and ideas of what constitutes a better system for itself, and how to go about assessing the developed system. The term "useful" is often used to describe a system, but without qualification, this does not tell anyone anything about the system. As seen earlier, Avison and Fitzgerald (1988)
suggest a list of generic qualities for analyzing the usefulness of an information system. It must be stressed that it is unlikely that any one system will satisfy all the criteria. However, each individual system can be adjusted to cater for each organizational need and requirement based on a combination of points such as acceptability, availability, cohesiveness, compatibility, documentation, ease of learning, economy, efficiency, fast development rate, feasibility, flexibility, functionality, implementability, low coupling, maintainability, portability, reliability, robustness, security, simplicity, testability, timeliness, usability, visibility.

The use of a methodology can help to control a development process. This means that management and project control can be improved by the use of an appropriate methodology. With an efficient and organised project management, productivity should improve as systems are built faster, given specific resources, or use fewer resources to achieve the same results.

Adopting a methodology as a standard approach to system development within an organization has the benefit of making the integration of sub-systems and future systems easier. This also means that the core of the development staff with their knowledge and experience can be utilized in other projects as well as maintenance.

The remainder of this chapter will discuss different trends in methodology and their associated methods.
2.2 Technically Oriented Methodologies

These methodologies concentrate on the technical aspects of proposed systems. This includes the hardware and software selection, the structure and method of data storage and retrieval, the security and integrity of the database.

2.2.1 Conventional System Analysis

This approach was first documented by the National Computing Centre in 1971 (Avison 1985) and has since been tried and tested. There are seven phases in the system development cycle. The feasibility study investigates the current problem, looking at alternative solutions and recommending the "best" solution. The systems investigation phase carries out a detailed investigation of the present system by interviewing, conducting questionnaires, sampling and observation. The systems analysis phase asks why the "problems" occurred, why the present methods were adopted and what the alternative methods are. In the systems design phase the new system is designed: clerical and computer procedures, data capture, file design, output design and security provisions. In the systems implementation phase the computer programs are designed, written and tested. Other activities include the training of users, the testing of clerical procedures, the production of documentation, the creation of master files and the eventual change from the old to the new system. The review and maintenance phase checks the running of the new system and makes changes as required. New requirements may be such that another new system is needed, and the project cycle starts again.

There are however problems with this method. Avison (1985) points out that a system produced by this approach may fail to meet the needs of users. This approach may also
produce systems that are inflexible, due to the constant changes in user environment and requirements. This inflexibility tends to lead to user dissatisfaction. There may also be problems with the documentation as it tends to be technically oriented. The system itself may be incomplete because only the basic design of the system was implemented. It may suffer from application backlog where parts of the system may not be built for some time after the original decision to proceed with the construction was made. Finally, with a poorly designed system comes the problem of maintenance. A lot of resources will have to be spent in keeping the system going. Avison (1985) concludes that all the problems mentioned above made the conventional approach to system design inadequate as a method for handling the increasing complexity of information systems.

2.2.2 Structured Systems Analysis

In the late 1970s, a more coordinated and systematic approach to system analysis was developed and promoted by many, including DeMarco, Yourdon (1979), Gane and Sarson (1979). Structured system analysis provides a more structured way of carrying out system analysis by using techniques and tools (Ashworth and Goodland 1990). In particular, it uses functional decomposition and stepwise refinement, which sets it apart from conventional methods. Structured analysis breaks down or decomposes a very complex problem into several less complex parts. Each part in turn is broken down into several parts. It structures a project into small, well-defined activities and specifies the sequence and interaction of these activities (Ashworth and Goodland 1990). Eventually a bottom level will be achieved, consisting of many easy and individually comprehensible parts. Avison (1985) argues that this leads to the production of reliable programs which, because of their
modular organization, are easier to maintain. Techniques used include data flow diagrams, data structure diagrams, decision trees, decision tables and structured English. By using diagrammatic and other modelling techniques will give a more precise definition that is understandable by both users and developers (Ashworth and Goodland 1990). The structured analysis documentation includes the logical analysis of the process as well as their physical level designs, the system's inputs, outputs, and data structures and processing logic.

Connor (1985) highlights the advantages of structured analysis and design. It is carried out step-by-step, each step handling manageable and effectively controlled pieces of information. The hierarchical design approach produces a well-organized system manageable by its users. The refinement of data flow diagrams produces modules which are independent and easy to maintain.

Ashworth and Goodland (1990) list the reasons for using structured methods. They point out that structured analysis provides a clear requirements statement that everyone can understand and is firm foundation for subsequent design and implementation. It helps to make more effective use of experienced and inexperienced staff. A structured method can spread experience more widely throughout the development teams, and certain tasks can be delegated to inexperienced staff who can be guided by the more experienced. Structured methods can also help with project planning and management. It breaks down a project into stages and steps and allows better estimation of time. It can also detect problems as they occur and not just before deadlines. Structured methods can produce high quality systems by making comprehensive specifications. Also the techniques used in the methods should be flexible enough to cope with changes. Together with formal quality assurance and informal walkthroughs at
the end of each stage, the analyst should be confident that the new system will meet user requirements before it is built.

However, there are also problems with the method. To use the various techniques, an understanding of their symbols, notations and operations is required. Anyone who uses them for the first time must learn these conventions, and their use in user meetings can lead to further problems due to lack of familiarization. The huge amount of data and information generated by the method can be difficult to manage. Any changes to the system specification can create problems, as the analyst teams must ensure all parts of the system affected are corrected in the light of the new specification. In structured analysis, when the present physical system description is evolved to the present logical system description, all time-oriented features of the system are removed until the proposed logical modules are assembled. By then it will need great skill and experience to put the time-oriented features back in. The conservation of data between levels is difficult, time-consuming, and is prone to errors even with the aid of data dictionary software.

However, even with all the problems mentioned above, the prospect of a system which meets the technical requirements and is easy to maintain in operational terms means that structured analysis and its derivative methods have been widely used. But system analysts and designers need to be well trained in the method and flexible in applying it.
2.2.2.1 SSADM

SSADM stands for Structured Systems Analysis and Design Method. It was developed by Learmonth and Burchett Management Systems in conjunction with the UK government's Central Computer and Telecommunication Agency (CCTA). The methodology was accepted by the CCTA in January 1981. In January 1983 it became compulsory for government departments to use SSADM for all systems development work. The method has been frequently updated with version 3 released in July 1986 and version 4 released in September 1989.

Downs et al (1988) recommends that SSADM is suitable for the development of medium to large systems. SSADM employs a "cook book" technique to help teams to apply the methodology. Each step in the methodology is described in detail: the tasks involved, the techniques used, the required input and the expected output.

It is argued (Downs et al 1988) that SSADM offers several design features which support the development of a system. Because SSADM is essentially data-driven, it uses techniques to capture the underlying and stable information handled within the system, how it flows into, out of, and around the system, and how the entities associated with the information change over time. SSADM requires that cross-checking be done at intervals to make sure that all areas of development work are done correctly and in a coherent manner. The logical and physical features of systems design are separated in SSADM. By doing this, the logical design can specify what an ideal information system would incorporate—e.g. a highly flexible, maintainable system. This will then be refined when the physical design is developed as a consequence of software availability, financial consideration, staff experience, and other physical
constraints. Therefore SSADM recommends that the transition from logical to physical design is done as late as possible.

SSADM does not provide any specific guidelines as to the composition of design and development groups who will be using the method for their work. It seems that within each design team, there has to be at least one expert SSADM user (possibly the systems analyst or an external consultant) and others with at least some knowledge of the method. The role of the management and staff of the organization throughout the development is not specified in SSADM and as such is a matter for design teams to decide.

In phase one, the feasibility study phase, the business case and technical feasibility of the project are considered. Estimated costs and benefits, and relevant social considerations are taken into account to see whether the analysis phase should go ahead. This phase is not compulsory but is recommended for a more thorough and proper use of the method. The two stages and their associated steps provide an "accurate" definition of the total problem to be addressed by the new system, and offer a set of options to tackle the problems defined in the previous stage, which are formalised into a feasibility report.

Phase two is the systems analysis phase, in which the current system, new user requirements and user problems with the current system are analyzed in more detail. This phase becomes important if the feasibility study phase is not carried out beforehand. If no feasibility study is carried out first, then analysts and design teams will be encountering the total problem for the first time at the stage where they should be concentrating on detailed analysis work. They will not have the benefits of conclusions from the feasibility study, such as whether the project is
actually viable, the definition of the problem to be addressed by the new system, and descriptions of the current system in the form of top level data flow diagrams. The overall project plan is another important document to have come out of the study. Without the study, all the groundwork for the project will have to be done at this phase, consuming more time and resource, and making this phase more significant.

The three stages and their associated steps carry out the analysis of system operations and current problems, and building on the user requirements expressed in terms of the problem/requirement list. Audit, security and control requirements are added in order to complete the required system specification. Users are invited, and assisted, later on in the last stage to choose a technical option for implementation of the required logical system, i.e. to select the required physical system.

The final phase is the systems design phase. It involves designing the required system both logically and physically using the specifications provided from the analysis phase. There are three stages in this phase: data design, process design and physical design. The data design stage involves the design of the data structures for the required system, culminating in the composite logical data design. This is often carried out in parallel with the process design. The process design stage is carried out in conjunction with the data design stage. In the physical design stage, the logical design from the previous two stages is used to build the required physical system. Once the detailed design work is accepted, the detail of the final system specification is created.
A detailed description of the phases, stages and steps of SSADM can be found in Appendix A.

2.2.2.2 The Use of SSADM in the UK

In 1987, members of staff from the Software Engineering Group of Sunderland Polytechnic's School of Computer Studies and Mathematics carried out a survey on the use of SSADM in projects in the public and private sectors of UK organizations (Edwards et al. 1989). This was part of the authors' goal to develop a useful, practical interface between SSADM and the Jackson Structured Programming (JSP), by facilitating the transition from the analysis and design stages to the software implementation phase. The main objectives of this survey were:

1) To discover how well users of SSADM felt the requirements and principles of the methodology have been fulfilled.

2) To determine specifically whether practitioners would welcome the development of a formalized interface between the SSADM and JSP methodologies.

3) To discover whether SSADM represents a strong force in the future of systems development.

4) To highlight areas for further research.

Three hundred and ten organizations were contacted and each was sent a questionnaire. The questionnaire was divided into the following sections:
1) General background information about the organization: the type and size of the organization; the hardware and software used.

2) Background information about the use of SSADM by the organization: the number of staff involved in SSADM projects; the number of SSADM projects attempted; the types of systems designed using SSADM; the training, support and commitment provided; the use of automated aids.

3) The techniques of SSADM: which ones are used and how they are valued by users and what other supporting, but non-SSADM techniques are used in projects.

4) The analysis and logical design stages of SSADM.

5) The physical design stage of SSADM.

6) The implementation of SSADM systems.

7) An addition section for constructive and relevant comments on areas that were considered not fully covered by the questionnaire.

There were 117 replies, 72 of which gave suitable data for analysis. The results of this survey provided much information about the areas mentioned above. But for the purpose of the following discussion, attention is focussed on the main conclusions of the survey.

The main survey finding suggested that the creation of a formalized interface between SSADM and the systems implementation phase would be an useful additional feature. Here is a summary of the other findings of the survey:—
1) SSADM provides guidelines and rules for the development of systems. It does not eliminate the need for good system analysts.

2) The analysis and logical design stages produce adequate logical design.

3) The physical design stage needs improved techniques, especially for physical design control, and further development in order to produce standard products for systems implementation.

4) The use of automated aids with SSADM will help the usage of the methodology.

5) Automated tools should be improved to provide more extensive support in the production and maintenance of SSADM products.

6) The implementation of systems using third generation languages, in particular COBOL, is still very common, despite the increasing talk of using fourth generation languages and application generators (mainly from the manufacturers of the related software and hardware).

7) The development of a practical interface between SSADM and systems implementation is a sensible goal.

2.2.3 Limitations in Structured Analysis Methods

In 1984, Nicholas Vitalari of the University of California carried out an assessment on structured analysis methods in terms of their ability to satisfactorily structure the problem solving behaviour of the systems analysts (Vitalari 1984). This assessment was based on a comparison
of structural analysis methods with research on the cognitive behaviour of systems analysts and the demands of the analysts task domain. Vitalari (1984) suggests that the major goal of most of these methods is to create a model of the information system having a high degree of internal and external validity. The assumptions are that the analysts know how to determine the requirements, and the final model will be the result of several iterations of model that have been developed together with users. Vitalari (1984) argues that none of these methods actually help in the requirements determination process. Instead they provide a way of specifying and documenting the requirements as a means of assuring that no obvious requirements are omitted. His assessment of structured analysis methods highlights ten limitations:

1) Current structured analysis methods are limited to a very small part of the requirements determination activities: specification, documentation, aiding the communication of requirements to users and other information systems personnel. They give little assistance to the initial structuring of the problem, setting goals, developing hypotheses, or formulating different strategies to determine information requirements. The structures provided by these techniques are mainly external and for the most part a framework for documentation and checks for internal consistency of the functional design.

2) The effectiveness of structured analysis methods depends on the level of experience of the system analysts using them. Experienced analysts can select from the various formalisms given in the methods and integrate them into their own problem solving process. Novice analysts however have not developed the additional skills of gathering information, limiting the size of the problem
space, and choosing the most satisfactory solution. For these novices, the techniques offer a narrow view of the analysis task domain. Therefore, when training system analysts to use structured analysis methods, other techniques have to be incorporated to bring in aspects of the analysis process such as goal setting, strategy formulation, hypotheses generation, and the realities of organizational structures and behaviour, politics and individual psychology.

3) Most of the current structured analysis methods are pre-scientific and non-empirical. It is not clear how the techniques should be integrated into analysts' approaches to problem solving. They are frequently asked to change their own approaches to the particular methods used. Research evidence gathered by Vitalari (1984) suggests that the techniques are best suited for application by analysts after they have actually completed the problem solving process and are ready to specify and document the requirements.

4) The various diagrammatic formalisms used in structured analysis methods may be useful for communicating the results amongst information systems personnel, but not so with the users. They tend to prefer a narrative approach or an active prototype of the system to understand the systems requirements.

5) Most of the techniques in structured analysis are developed to model the flows and structure of transaction processing systems that have distinct packets of data and well defined files. They are less useful with other systems such as decision support, electronic spreadsheet applications, systems that use a great deal of internal processing and derived data such as tax projection and
investment tax credit analysis systems. In these task domains the techniques become restrictive and interfere with the analyst’s problem solving process.

6) Structured analysis methods give the impression that the analysis activity is a step-by-step, sequential process which, if properly carried out, will result in high quality requirements. Research evidence carried on the problem solving process, gathered by Vitalari (1984), shows that it is iterative and non-procedural. Because of the complexity and dynamic nature of analysis problems, analysts should adopt a more flexible and non-sequential approach.

7) Structured analysis methods do not help in problem planning. They provide guidelines for documentation and rules for assessing internal validity. But the analysts themselves have to decide what the problems are and how to go about solving them.

8) There is uncertainty on how well present structured analysis methods can adapt to and evolve with the development technology in the future, such as rapid prototyping, end-user programming, and automated systems development techniques.

9) Structured analysis methods give analysts a restricted view of the context in which systems analysis occurs. The modelling techniques are abstract and are difficult to use for documenting changes in the organizational structure.
10) **Structured analysis methods provide little support and guidance for the analyst to assist problem diagnosis and determination.**

2.2.4 **Summary**

Structured analysis methods have been developing for over twenty years, supported by prominent advocates such as DeMarco, Yourdon, and Gane and Sarson. SSADM is one such method which has been used in projects in public and private organizations. Structured analysis methods break down a complex problem into smaller and manageable parts to be solved. A variety of techniques are used to help to solve these problems. SSADM is a prescriptive formalization of the methodology. At each step, the techniques required are specified, as well as the inputs necessary and the outputs expected. A survey carried out by Sunderland Polytechnic indicates that SSADM is widely used by many organizations. Replies to the survey suggest improvements to the use of SSADM. A further study by Vitalari indicates that structured analysis is good in dealing with requirements specifications, but offers little help in problem definition, diagnosis, formulation, planning, which are all part of systems design. This in turn means a heavy reliance on the experience and knowledge of analysts, and their skills in applying the method to their specific problems.
2.3 Socially Oriented Methodologies

The other major trend in system analysis and design is the socially oriented methodologies and their methods. These methodologies concentrate on the social implications and organizational impact of new systems on a business. Their main emphasis is on the requirements and aspirations of the staff and the jobs they do during analysis, and thus on designing a system that will meet these demands.

2.3.1 Participative Design

Professor Enid Mumford of the Manchester Business School (Mumford 1986) describes participative design as a practical approach which allows users to participate in the analysis and design of the system they will be using, thus making the implementation, acceptance and operation of the system easier. To achieve this end requires the involvement of everyone from top management to departmental staff. In participative design the role of the analyst is more of a facilitator advising on the possibilities from which the user chooses.

There are three approaches to participation. The consultative approach leaves the main design tasks to the analysts who consult with the users about any required changes. Most conventional system development methods require this level of user involvement. The representative approach involves the setting up of a design group, consisting of user representatives (selected by management) and the analysts. Everyone in the group is on equal terms and has an equal say in the design process. However they have to truly represent the interests of those who are affected. The consensus approach tries to involve all user staff continuously throughout the design process. The user
representatives in the design group are elected by the staff of all the departments. This is supposed to allow ideas and reactions to flow between the design team and the departments. However this democracy also means that it is more difficult to make decisions.

There are a number of problems with this design approach (Humford 1986). A significant and sustainable amount of trust between the management and the staff is needed throughout the design process. Also there is the potential for dispute between the elected design group members and the selected members. In addition, conflicts of interest within the design group will appear very quickly. As a result, stress occurs which can adversely affect progress. This highlights the importance of good communication and consultation in a design group. Such skills are not easily obtained. Also, the professional system designer, by becoming more of a consultant, loses some of his authority and status. Departmental managers need to be kept in touch with what is taking place if they are not actual members of the design group, as they have to agree on decisions made by their staff. The facilitator in the group has to keep the morale high as it inevitably fluctuates throughout the project.

With these problems in mind, the participative approach takes up the issue of conflicts with changes, and tackles it by bringing it into the open to be discussed, negotiated and resolved by all parties concerned. It is viewed as pragmatic, ideologically and morally right, which helps efficiency, satisfaction and progress.
2.3.1.1 ETHICS

ETHICS stands for Effective Technical and Human Implementation of Computer-based Systems. This is a participative design approach developed by Humford (Humford 1986). ETHICS is used to ensure that the new system is valuable to the organization. As all levels of the organization are involved in its design, the system will be "their baby to bring up and look after". Systems designed using ETHICS should meet the job satisfaction objectives in addition to the technical and operational ones, and will be surrounded by a compatible, well functioning organizational system.

At the design and development level, ETHICS is used because:

1) It bases system design on an accurate and careful diagnosis of business problems and human relations needs.

2) It gives equal weight to these problems and needs.

3) It ensures that the design task encompasses good organizational design as well as good technical design.

4) It creates systems which are effective, efficient, acceptable and stimulating.

An aim of ETHICS is to involve those people who will ultimately be using the new system in the whole design process. Members of staff, management and the professional system analysts are all involved in the process, from the initiation of the project up to its implementation and evaluation.
The ETHICS process starts with the setting up of two working groups. The Steering Committee consists of senior management from the user, systems development, finance and all other major areas, including senior union officials where necessary. It sets guidelines for the Design Group. The Design Group consists of representatives of all those interested in the design area as well as the professional systems analysts.

A detailed description of ETHICS can be found in Appendix B.

2.3.2 Experience with Participative Design

In 1976, ETHICS was used by the Derby Engine Group of Rolls-Royce Limited (Mumford and Henshall, 1979). Mumford was asked to assist the company in the introduction of a series of on-line computer systems into the accountancy functions of the Treasurer's Department and the Purchase Invoice Department. The company was already an experienced user of computers, and it saw this as an opportunity to carry out an office reorganization to improve both job satisfaction and efficiency. The system would control the Bought Ledger and mechanize the cheque issue and remittance routines. Its on-line nature could provide a useful service for the Purchase Invoice Department through assisting them to answer queries and trace documents more quickly. The project's history was published by Mumford and the company's Systems Manager, Don Henshall (Mumford and Henshall 1979), in a book in which conclusions and comments about the use of the participative approach to systems design were documented.

One of the first acts carried out during the project was to secure a guarantee from both the management and the trade unions that there would be no redundancy, and that any
staff saving would be achieved by natural wastage, or by redeployment if necessary. The proposed system would give most of the clerks the opportunity to increase the range of tasks they could perform. This would in turn help their promotion prospects to higher job grades and increased salary. Those senior clerks who were already at the top would assume new responsibilities of training and co-ordinating junior staff within the new organization.

There was no formal negotiation between management and the clerks. The project teams consisted of many of the clerks who would ultimately use the systems. Therefore the right personnel were making the appropriate decisions. However, in order for these people to make the decisions, a lot of informal negotiation was required between the clerks in the design groups and their colleagues. The trades union APEX (Association of Professional Executive, Clerical and Computer Staff) also gave assistance and encouragement to the Design Group.

One of the difficult problems was the degree of involvement of the different levels of management in the company. The front line management, such as the section leaders, were involved in the design process and were represented in the design group. The departmental supervisors were consulted and kept informed during the process. The middle management was however difficult to deal with. Too much involvement from them would inhibit the design groups and would tend to cause them to move too close towards management ideas. Too little involvement from middle management would mean that the new organization could never be successfully implemented. Therefore a balanced approach was needed so that the middle management was involved to a suitable extent. The senior management were involved in the Steering Group so that everyone involved in the project would
know that they had the support from them and could therefore persevere through the difficult phases of the design process.

The design group had to transform itself from a group of clerks whose knowledge was restricted to clerical operation, to a group capable of creative problem-solving and the generation of new, radical ideas. The departments too had to acquire skills in identifying work problems and suggested improvements. They had to be able to communicate these ideas to the design groups. A major difficulty was that the clerks were not convinced that the management was asking them to design the system. The clerks felt that the management could have asked professional designers to build the system, and were therefore suspicious of the motives of the management. Another difficulty was the need to change from designing the technical system and then reorganising the work system to fit the system, to a more flexible approach to the technical design process that avoided imposing constraints on the Design Group by insisting on a particular technical solution. All these matters meant that the training of the system analysts had to include the appropriate skills and techniques to cope with them.

The only financial cost involved was the employment of the external consultant (Mumford). The introduction of the new system was carried out by the clerks themselves and not the system analysts. In practice the work of the departments was not badly affected by the absence of five clerks for half a day per week.

Mumford and Henshall (1979) set out the argument for the participative approach: "All change involves some conflicts of interest. To be resolved, these conflicts need to be recognised, brought out into the open, negotiated and a solution arrived at which largely meets the interests of all
parties in the situation. Differences of interests will not be confined to management and subordinates but will occur between employees at different hierarchical levels, and in different functions. Therefore successful change strategies require institutional mechanisms which enable all these interests to be represented, and the participative design group, which consists of representatives of all the different groups in a department, will fulfill this function. An excellent relationship between members of the representative design group and their constituencies helps to carry out this approach.

Humford and Henshall (1979) also point out that many managers, employees and trades unions who believe strongly in the ethic of participation, would argue that those who will ultimately be affected by change should have the chance to determine their own destiny. Otherwise, they might feel themselves victims of the thoughtlessness, lack of imagination, or ill will of the creators of change. Humford and Henshall further suggest that neither management nor their subordinates gain from a purely instrumental relationship to work in which the employee gives his or her labour in exchange for money, with financial gain being the major source of satisfaction. This is undesirable to the employee as it does not stimulate the development of his talents and the realization of his potential as a person. It is also undesirable to the company as it produces a set of individuals with little motivation to do anything other than exchange their labour for cash. Once the cash is in hand, there is no stimulation to work harder or achieve more. Therefore the company's growth and success will be held back.
2.4 Other Methodologies

So far, this chapter has described and discussed methodologies which are technically and socially oriented. However, there are other emerging methodologies which are based on other aspects and different philosophies all together. In the next section we consider Checkland's Soft Systems Methodology (1981) and the increasingly popular concept of prototyping (Harker 1988).

2.4.1 Soft Systems Methodology

This methodology was developed after extensive research carried out by, amongst others, Checkland (1981). It aims to provide an alternative to the hard systems methodology. Hard systems methodology tackles real-world problems in which an objective or end-to-be-achieved can be taken as given, and a system is then engineered to achieve the stated objective. Soft systems methodology tackles real-world problems in which known-to-be-desirable ends cannot be taken as given. The methodology has four characteristics: it is capable of being used in actual problem situations; it is not vague in the sense that it provides a greater spur to action than a general everyday philosophy; it is not precise, but allows insights which precision might exclude; any development in system science can be included in the methodology and can be used if appropriate in a particular situation. It consists of two kinds of activity: real-world activities involving people in the problem situation; "systems thinking" activities which may not involve those in the problem situation. There are seven stages in the methodology, in which stages 1, 2, 5, 6 and 7 are the real-world activities, and the remaining stages are the "system thinking" activities.
Stages 1 and 2 are an expression phase during which the most detailed picture is constructed of the situation in which there is perceived to be a problem. This initial analysis is carried out by recording elements of slow-to-change structure within the situation and elements of continuously-changing process, and forming a view of how structure and process relate to each other within the situation being investigated.

Stage 3, known as root definition, puts forward systems which may be relevant to the supposed problem by preparing concise definitions of what these systems are. The purpose is to obtain careful and precise descriptions of the nature of the systems so that the fundamental nature of the systems chosen is captured to assist future decision making.

Stage 4 builds conceptual models of the systems defined in the root definition. A structured set of verbs is used to describe the minimum necessary activities required by the systems. There are two parts in this step. The first step is the use of a general model of any human activity system which can be used to check that the models built are not fundamentally deficient. The second step is to modify or transform the model into any other form which may be considered suitable in a particular problem.

Stage 5 brings the models built into the real world and sets them against perceptions of what exists there. The purpose is to generate debate amongst the people concerned. The root definitions and conceptual models are compared with the perceived realities in the problem situation. There are four different ways of carrying out the comparison:--
1) Ordered questioning based on the conceptual model;

2) Reconstructing a sequence of events in the past and comparing what had happening during production with what would have happened if the relevant conceptual models had actually been implemented;

3) A more general comparison, asking what features of the conceptual models are especially different from the present realities and why;

4) Building a second model of what exists, and directly overlaying this onto the conceptual model with the aim of revealing mismatches for discussion.

These methods of comparison help to ensure that it is done in a conscious, coherent and defensible manner.

Stage 6 uses the comparison done in stage 5 to generate discussion on the changes in structure, procedures and attitudes. The discussion should be with the people concerned with the new system. The changes defined should be desirable and feasible.

Stage 7 implements the changes discussed in the previous stage.

2.4.2 Prototyping

The successful development of any system that meets user requirements depends on the design of the interface which supports the match with the tasks of individual end user and the organization (Harker 1988). In a large complex system, problems with interface design can be difficult to resolve as these affect the nature of the system itself as
well as a huge number of people. Therefore, user involvement from the beginning of the interface design is important to ensure that an interface which meets user requirements. However, it is also important that these user requirements are effectively expressed by the users, and captured by the analysts. Users often find the analysis methods difficult to understand, and the analysts find user description not precise enough to be used as the basis for the design specifications.

One of the solutions to this communication problems is the use prototypes and simulations to represent information from both sides. There is little distinction between prototyping and simulation (Harker 1988), but in general for large systems it would be better to simulate the interface and the internal information processing, rather than just prototyping the technical aspects and observing user reactions. There are two other solutions to the knowledge elicitation problems. One is demonstration whereby users are shown a similar system at work on site. However, this may not be suitable as such a system may not be developed from requirements specific to users. The other is the development of a pilot system to handle a realistic working conditions in a limited scale. However, such a pilot system should be developed and evaluated well before the main system is finalised. Otherwise, any outcome from the pilot system will unlikely to have any effect on the decision on the main system. Developing a pilot system takes time, which may not be available. Prototyping and simulation are useful alternatives to the ones mentioned above. For the purpose of this part of the thesis, prototyping will be the area of interest.

Prototyping should be used at the early stages of systems design process, such as the initiation, specification
and logical design. By using it at these stages, user issues can be dealt with during specification of the system. Aspects of the system which have the most impact on the users should be highlighted so that testing can be carried out to obtain useful information to the designers. In order to use prototyping effectively, assumptions should be made regarding the nature of the system for which the technical support will be needed. The prototype should then be developed according to these assumptions, incorporating the user interface characteristics. The developed prototype should then be given to the user to try out to gain some hands-on experience. Harker (1988) points out that letting users interact with the prototype will provide far better judgement on its strengths and weaknesses because of the dynamic interactions generated, and realistic user reactions. Harker also points out that in order for users to give realistic reactions, the tasks performed by the prototype should also be realistic with any input information similar to that which will be going into the full system. Sufficient time should also be given for the users to try out the prototype in order to assess the acceptability of the prototype. Task duration ranges from under a minute to several hours or even days. Therefore, to gain realistic and useful information, a decision should be made on the complexity and size of the prototype with trial time consideration in mind.

There are three types of prototyping (Kingston 1987). The "throw-away" type is often constructed quickly to test a small part of the system, such as dialogue design and screen layout. It is normally discarded afterwards and is hardly documented. The incremental prototype is constructed more carefully. After each demonstration of the prototype, any changes are incorporated. The next stage of the development is then built on top of the prototype. This process continues until the eventual system is completed. Each
prototype system is carefully documented to show the users what they have agreed upon and how the system is developing. The evolutionary prototype is similar to the incremental one. But after each demonstration of the prototype, changes are made, documented and a copy of this is stored away. Then the next stage of development begins. In this way, users can see how the system is evolving into the proposed system.

Prototyping can reduce the application development time and cost in the long term (1987), thus avoiding the problem of rejection after implementation. However, clients may have doubts over the time and money spent on the construction of the prototype. After carrying out a comparison between use of paper mockup and computer mockup (prototyping), Nielsen (1990) came to the following conclusion: computer mockups focus on the major usability problems and offer the evaluator an experience close to end user; paper mockups are better at showing certain inconsistencies in a design.

2.5 Combination Methodologies

Recently, there is a trend away from relying on one single methodology to carry out system analysis and design, and a move towards methodologies which combine techniques and procedures of other methodologies or methods in a coordinated and complimentary way to produce a more complete design methodology. This section will cover three of them: DIADEM, recently used by a large UK organization (Ip et al 1990); the user-centred method (Eason 1988), and Wood-Harper's Multiview (Wood-Harper et al 1985).
2.5.1 DIADEM

DIADEM stands for the Departmental Integrated Application Development Methodology. This method is used by a large well established UK organization as its system development methodology. This methodology provides procedures which incorporate human factors consideration so that the eventual system produced will be more usable and acceptable to the user (Ip et al 1990). The DIADEM documentation states that the methodology integrates the principles of good project management with the tools and techniques of SSADM. It also states that the overall role of the methodology is to provide a framework within which systems development activities may proceed in an organized and co-ordinated manner.

The following is a brief summary of the techniques used and the user involvement in the first two stages of the method: the initiation stage and the specification and logical design stage (DIADEM 1987). A detailed description of the methodology can be found in Appendix C.

In the initiation stage, a Project Steering Committee is appointed, consisting of senior management from the user branch, systems development, finance and all other major interfaces. A user Committee is also appointed, consisting of users not working on the project team. The chairman of this committee should be a senior user who is also on the steering committee. The purpose of these committees is to establish project plans, to set up project milestones, to organise resource and to agree on standards.

Following this the existing system is surveyed. The techniques used include data flow diagrams, logical data structuring, job stream charting, and risk assessment. These
require detailed interviews with key personnel and/or questionnaires. Problems of the current system are listed and proposals for new requirements are made. The current problems are analyzed with the user and agreement is reached on the changes needed and the impact of the changes on the system environment, staff, etc. The ideal current data flow diagrams and logical data structures are then produced, followed by the required data flow diagrams and logical data structures. The next step is to outline the functional requirements using job stream charting and dialogue design. This is followed by identifying user acceptance criteria. Finally, the organizational impact of the proposed system is assessed. This requires full user involvement as recommendations for job design, grading and staffing level are made, together with the user support requirement during development and implementation.

In the specification and logical design stage, all the results of the current system survey are reviewed with the user. Any changes must be re-confirmed explicitly with them and provisional logical data structures drawn up. During the review, aspects of dissatisfaction, organizational problems, and urgency of problem resolution are collected. All the current physical and logical data flow diagrams and logical data structures are also reviewed with the user, ensuring that they understand the symbols, notations, and meanings of the diagrams. Any new changes agreed should be incorporated into the diagrams. Other areas in the survey to be reviewed, and be updated if necessary, include the user-computer interfaces, system objectives and task objectives. After the current system survey review is completed, the user specification is produced. Then a formal user review of the work done so far is carried out. This involves a presentation, followed by a question-and-answer session afterwards. All key personnel of the user group attend,
together with those in the project team. The project will not progress further if there are still major problems or disagreements.

A discussion on the experience of using DIADEM will be in a later Chapter. The combination of aspects of human factors techniques into SSADM had mixed results when the methodology was used in the project. But the underlying principles of the integration offers an effective way of improving the use of SSADM.

2.5.2 User-Centred Method

The user-centred method has been promoted by many recently (Harker & Eason 1984; Eason 1988). The method came about from the study of system analysis and design methods. Eason (1988) concluded that current methods tend to only emphasize part of the functional aspects of a system. Secondly, although some methods do address non-technical issues, they do not provide the technique or the expertise to cope with them. The Human Science and Advanced Technology (HUSAT) Research Institute at Loughborough University has provided some guidelines on what a good method of analysis and design should be. It should include user considerations, individual, social and organizational needs. It should be a combination of top-down and bottom-up approaches. However it should still maintain some technical orientation. This user-centred approach should be used as a set of techniques within a set of design methodologies. The following are the techniques used in the first two phases of the user-centred technique: the feasibility phase and the requirements specification phase.

In the feasibility phase, the socio-technical analysis is used to identify the major problems and opportunities
facing an organization. The cost-benefit assessments of organizational impact is also used to estimate the likely effects and implications of the proposed system to the user organization. With the proposed system beginning to take shape, the composition of the design team can now be considered, including both user and technical members.

In the requirements specification phase, the organizational realities of the user environment should be translated into the socio-technical requirements. A suggested method is the Open System Task Analysis technique. It is also important to establish the organizational forms that are required before defining any technical aspect of the proposed system. As for the specification of the technical system, the user should come up with suggestions on what they would like, and the various interfaces and support requirements. To help the user to achieve this, the evaluation of an early prototype can help the user to understand what the proposed system may look like.

Involving all potential users in the design of a system is a good way of producing a usable end system, but there are problems with using this particular method (Eason 1988). The first is the number of users that will be involved. In an organization with a small number of end users, this problem may not be too serious, but still a major one. It is simply implausible to involve every single end user in every aspect of system design. It will be difficult to organize, coordinate, discuss, access, and incorporate every single idea from the end users. This problem can be resolved by adopting the minimum critical specification strategy. This allows the designers to fix the design of the system up to a certain point, and leave matters such as job design, technical system service, office layout, and implementation procedure to working groups of local users.
The second problem is the knowledge the users require to take part in the design process. This is similar to that faced in the participative approach, but this time it is not only of concern to the user representatives but to everyone. This problem can be partly resolved by carefully assigning people to work on areas which they have knowledge of, and to create ways to help others to acquire the knowledge necessary to be involved in decision making process.

The third problem essentially concerns the management of a project using the user-centred method. With so many people involved in the development process, the management of the organization will be worried that different parts of the system developed cannot be put together into a fully operational one. Technical and maintenance staff will be concerned that the final system will be serving different and conflicting objectives and difficult to maintain. The solution will have to come from an effective project management and steering group to oversee the development of the system.

These problems occur when the balance shifts too far towards the end user requirements and away from the management and technical goals. Therefore, the project management needs to establish the terms of reference, composition and time scales of sub-groups charged with parts of the process. It will also have to make clear that participation of all users does not mean that everything is possible but there are many choices within limits.
2.5.3 Multiview

This method was developed by Wood-Harper, Antill and Avison (1985). It combines important aspects of several trends of system design: human activity systems (Checkland), socio-technical systems (Mumford), data analysis and structured analysis (deMarco, Gane and Sarson). There are five stages in the methodology, which move from the general to the specific, the conceptual to hard fact, and from issues to tasks. They should be followed in this order: the analysis of human activity systems, the analysis of entities and functions, the analysis and design of socio-technical systems, the design of human-computer interaction, the design of technical subsystems.

Stage one is the analysis of human activity systems. This stage is essentially the work of Checkland. The central focus of this stage is to search for particular views which form the basis for describing the systems requirements. The search is in the form of debates on the main purpose of the organization concerned. The analyst or the project team forms a rich picture of the problem situation with the help of those for whom the analysis is done. This picture is both subjective and objective, consisting of the clients of the system, the people taking a part in it, the tasks being performed, the environment, and the owner of the system. It can be used to generate and to take part in the debate. From the rich picture the analyst extracts problem themes and puts forward relevant systems that can solve them. Several systems should be suggested in order to assist in the debate. Once a view or root definition has been decided upon, it can then be refined and developed. When this satisfies both the analyst and the users, then a conceptual model of the system is constructed. This is done by compiling a list of verbs covering the necessary activities in the system defined in
the root definition. By now, we will have a description of what the system will be and an inference diagram of the activities the system will carry out. This conceptual model is then compared to the rich picture constructed earlier. Any differences are noted and discussed. Any possible changes are debated before being implemented.

Stage 2 is the analysis of entities and functions, or information modelling. This is done independently from any consideration of how the system will eventually develop. The major input to this stage is the root definition/conceptual model from the stage 1. There are two phases involved in this stage. The first is the functional decomposition when the main functions of the system are identified and then broken down progressively into subfunctions, until the level is reached when the analyst decides that further breakdown is not useful. Data flow diagrams are constructed to show the breakdown. The second phase is the construction of the entity model. The analyst identifies entities and establishes relationships between them. Work done in stage 1 in understanding the problem situation helps to carry out this process. An entity model is then constructed, and further refined to be used as input into later stages.

Stage 3 is the analysis and design of the socio-technical system, based on the work of Humford. The main concern here is the identification of alternative arrangements to the social and technical objectives. They are then combined to produce the socio-technical alternatives. These are ranked in terms of their fulfillment of the previous sets of objectives, costs, resources and constraints of each objective. In the end the best socio-technical solution is selected and the corresponding computer tasks, role sets and people tasks defined. The importance of this stage is to produce a statement of alternative systems.
and choose between them according to important social and technical considerations. The outputs of this stage are the computer task requirements, the role set, the people tasks, and the social subsystem. These are also major outputs of Multiview.

Stage 4 is the design of the human-computer interface. The inputs to this stage are the entity model from stage 2, the computer tasks, role set and people tasks from stage 3. The stage concerns the technical design of the human-computer interface, and makes specific decisions on the technical system alternatives. Once the interface is defined, the technical requirements to achieve this can be designed. These technical requirements are used in the final stage. The interface definition itself becomes a major output from Multiview.

Stage 5 is the design of the technical subsystem. The major inputs here are the entity model and the technical requirements. Since the necessary human consideration, social and technical objectives have been taken into account to produce the input products, the analyst can concentrate on the efficient design and production of the full systems specification. Many technical criteria are analyzed and technical decisions made which takes into account all the previous analysis and design stages. The final major outputs are the application subsystems and the non-application subsystems. Such subsystems include information retrieval, control, database, database maintenance and recovery. The inputs and outputs necessary to support these subsystems are also defined.

To summarise, the final outputs of Multiview are the socio-technical systems, the role set and people tasks, the human-computer interface, the technical specification, and
the necessary inputs and outputs to support the non-application subsystems. These it is claimed (1985) include all the necessary information to design, implement, operate and maintain a more complete system in both human and technical terms.

2.6 Summary

In this Chapter, the need for a methodology for system analysis and design was discussed. The three reasons given were that by using a methodology one should have a better end product, a better development process, and a standardized process for future use. Then the technically and socially oriented systems design methodologies were discussed. This was followed by other methodologies which take alternative approaches to systems design, as well as combining different methods. The various methodologies and methods derived from them were also discussed.

In the technical section, conventional analysis was discussed in which a development cycle technique was used. But because of its inflexibility and user dissatisfaction, structured analysis was then developed to provide a more coordinated and systematic approach. In particular, SSADM has been highlighted because of its adoption as the UK Government's standard method for its commissions. A report of its use by the UK government and commercial projects was discussed, together with a study of the limitations of structured analysis. Even with these limitations, structured analysis methods are still the most widely used at present. They concentrate on system construction, which can be seen in terms of the hardware and software the system is using. They are used to illustrate the success or failure of a system. The social and human aspects of the system are hardly
featured in structured analysis as they are less straightforward to elicit from end users, and therefore tend to be left out from the final consideration during structured analysis.

In the social methods section, participative design was discussed whereby the users were heavily involved in the design process. Mumford's ETHICS method was described and a report of its use in Rolls-Royce highlighted, where it showed how the problems of trust between management and staff, communication and consultation were resolved by the company. Social methods involve some form of exploration into individual needs and aspirations. Some of the staff will indeed have their own views on what the proposed system should do. However, not everyone who eventually uses the final system will necessarily have particular demands and requirements when the system construction is started. If this situation extends to many of the end users, then the ideals, objectives and operations of the social methods will be undermined.

Other methodologies mentioned included Checkland's Soft Systems Methodology, in which a practical analysis process is combined with a philosophical approach to analyze what the needs and requirements of the new system would be. Prototyping was also discussed as a way to show users the different alternatives to the proposed system without incurring much cost.

The combination methodologies discussed included: DIADEM which incorporates human factors in SSADM; the user-
centred method in which user considerations, individual, social and organizational needs are included in the technical design process: Multiview which combines Checkland's and Mumford's methods together with data and structured analysis. These combination methodologies try to link up different aspects of several methodologies in order to form a more all round one. The present trend seems to be to try to incorporate techniques from a social method into a technical method. This means that the technical method is still the dominant one, with some additional improvements to it.

The discussion so far in this Chapter has been primarily about theoretical methods of analysis. It is essential to see how these work in practice. Most of the currently favoured methods employed in practice are mainly technically oriented. This may be due to the fact that these methods tend to be instructional and prescriptive. Other methodologies require consideration of social or human requirements, as well as an understanding of their philosophical approaches to the design process. The following Chapter will look at attempts at comparisons as well as highlighting a comparative study carried out recently on several technical methods.
3. Methodology Comparison

In this chapter, the problems in comparing different methodologies will be discussed, followed by a suggested framework for comparison. Then the findings of a comparison exercise carried by the Technical University of Berlin from 1982 to 1985 will be highlighted, followed by some suggestions for incorporating social and human factors techniques into the technical methods, based on the finding of the Berlin study.

3.1 Problems with Comparison

Before highlighting the problems with comparing methodologies, it is worth considering why one will want to compare at all. There are two main reasons for comparing methodologies (Avison & Fitzgerald 1988). The first one is an academic comparison to understand the nature of methodologies, such as their features, objectives and philosophies, in order to clearly identify each methodology and to improve future methodologies. The second reason is a practical comparison in order to choose a methodology, part of one, or a number of them for a particular application, a group of applications, or as a standard for an organization.

There are problems in carrying out a proper comparison of the various methodologies. A serious attempt was made by the International Federation of Information Processing (IFIP). They set up a working group which organized a series of conferences known as the Comparative Review of Information Systems Design Methodologies (CRIS). They started work in 1982 and by 1988 had held several conferences and published their findings. The work of the group was recognised to have made important contributions to the field but it also
attracted criticisms. Avison & Fitzgerald (1988) highlighted four main ones:–

1) Many of the methodologies touched upon by the working group had been theoretical ones with very few practical users. This implied that the whole comparison exercise might have become purely academic.

2) When the case study was selected as the basis for comparison, a detailed user requirements specification already existed. This meant that no early analysis work was required at all. This would also restrict the applicability of the results in respect of problems where analysis is an important and integral part of the solution process.

3) During the comparison, the feature analysis part caused problems. Some of the originators and authors of methodologies argued that the feature analysis did not correctly understand and interpret their particular methodologies. This disagreement undermines the validity of the findings because of uncertainties concerning the criteria for comparison.

4) There were further problems with the feature analysis. It did not take into account the practicalities in the application of the methodologies in the commercial environment. The analysis did not evaluate the likely success of particular features, it merely confirmed their existence in the various methodologies.

Apart from the problems of this particular attempt, there are inherent difficulties in comparison. Checkland (1987) points out that it is impossible to prove that the use of a methodology contributes to the success or failure of the
system. Checkland argues that system developers cannot claim that by using a different methodology that they could not have done the job better. He also argues that system developers cannot refuse to accept that problems during development can be due to their own failings rather than the methodology itself.

Another inherent problem is the evolutionary nature of methodologies. The area of information systems is still developing, and new technical advancements cause methodologies to change. Several consequences follow from this. The documentation of the methodology changes, and therefore the problem of knowing which version of the methodology is in use and whether the correct documentation is at hand. The documentation may not necessarily be available to non-fully subscribed to, or paid up users of, the methodology. This may mean that the practice of the methodology may differ from what is actually documented. Finally, individuals or groups of developers who use the methodology will interpret its usage differently from one another.

A further problem involves the use of terminology. Information system methodologies are notorious for the many different terms used for the same phenomena, and similar terms that actually refer to different phenomena. This leads to confusion and poor communication. This also inhibits the evolution and development of the methodology. Ultimately, this may lead to the methodology being ignored or even rejected.
3.2 Methods of Comparison

3.2.1 Feature Analysis

The most common method of comparing different "objects", although not always recognised, is to identify a set of features considered to be important. The objects are then checked against this features list. Those which possess most of these features are deemed to be a good match, whereas those which do not are said to be less of a good match. Information system methodologies can be compared in this way as well. However, the set of features may be compiled subjectively by interested or biased parties, with the motive of ensuring their own product appears favorably in the comparison.

3.2.1.1 Maddison

An alternative way is to allow the user to put down his own scores throughout the features list. This was suggested by Maddison in 1983 (Avison & Fitzgerald 1988), and he himself suggested a features list of over 100 items. Although these were his own suggestions, and therefore still subjective, the list is more generic and non-product oriented. They were summarised by Avison and Fitzgerald (1988) to the following major points:

1) Does the methodology cover all aspects of the systems analysis and design process from planning to implementation?
2) Are the steps well defined?
3) Is the methodology data or process oriented?
4) How are the results at each stage expressed?
5) To what type of applications is it suited?
6) Does it aim to be scientific or behavioral?
7) Is a computer solution assumed? What are the other assumptions made?
8) Who plays the major role, the analyst or the user?
9) What built-in controls are there to evaluate the success of each stage?
10) Is the methodology simply an attempt to link a number of techniques and tools or does it have its own philosophical base?

The last point was stressed by Haddison (Avison & Fitzgerald 1988) as very important. A clear understanding of the philosophy of the methodology will mean that a major obstacle to the usage of the methodology has been overcome.

3.2.1.2 Catchpole

A lot of effort has been put into identifying the important areas on which comparison between methodologies ought to be based, and which should be included in any future design methodology. Catchpole (1987) collated them and his work was summarized by Avison and Fitzgerald (1988).

Catchpole collated his requirements from authors including Bantleman, Land, Macdonald, Tozer, Wasserman and Freeman, Wood-Harper and Yao. The following are the requirements list in Catchpole's work (they are not in any order): guidelines to its use, total coverage, understanding the information source, documentation standards, separation of logical and physical designs, validity of design, early change, inter-stage communication, effective problem analysis, planning and control, performance evaluation, increased productivity, improved quality, visibility of the product, teachable, information system boundary, designing for change, effective communication, simplicity, ongoing relevance, automated development aids, consideration of user...
goals and objectives, participation, relevance to practitioner, relevance to application, a systematic way of looking into the future, integration of the technical and the non-technical systems, scan for opportunity, separation of analysis and design.

3.2.2 Alternatives to Feature Analysis

There are several alternatives other than feature analysis which can be used as a basis for comparing methodologies, and a summary of the main points of each method is given here.

3.2.2.1 Bubenko

Bubenko proposed three areas for comparison. The first is a theoretical investigation of well defined, narrow, subject areas of the methodology within specific terms of reference. It is hoped that this will decrease the problems of subjectivity. However, in order to understand the investigation, it has to be carried out in an unambiguous way. This means that the outcome of such an investigation will be a very formal one as this is one of the only ways of avoiding ambiguous statements. This formality means that the use of natural languages is not suitable.

The second is the case study approach. This means the gathering of experiences in applying the methodology to real cases. The objectives of the study have to be fully explicit and adhered to in the performance of the study. It is very difficult to take into account environment factors, and as mentioned earlier there is the problem of the proof of success argued by Checkland (1987). Finding the right cases for comparison is yet another difficult problem.
The third area for comparison is cognitive investigations. This examines the way in which software psychology and human computer interaction are handled and incorporated into the methodology. One of the questions this seeks to answer is whether graphical techniques give better understanding than formal languages. Another question is the meaning of "understanding" itself. This third area is probably the most difficult to compare of Bubenko's suggestions.

3.2.2.2 Davis

(Avison & Fitzgerald, 1988)

Davis advocates the contingency approach where the emphasis is on the search for an appropriate methodology in the context of the problems being addressed, the applications, the organization and its culture. He suggests that the level of uncertainty in a system should be measured, and there are four of them: the system complexity or ill-structuredness, the state of flux of the system, the user component of the system, the level of skill and experience of the analysts. Once the level is established, the appropriate approach can be determined. For example, a low uncertainty level suggest that the traditional method of interviewing users would be sufficient. A high uncertainty level would mean using prototyping or an evolutionary approach. An intermediate level of uncertainty would mean using another approach.

3.2.2.3 Episkopou and Wood-Harper

(Avison & Fitzgerald, 1988)

Episkopou and Wood-Harper suggest that a suitable approach should be determined by examining variables within and around the problem situation. There are three areas of concern: the problem content system where the problem, its environment, and its owners are contained; the
problem solving system where the methodology or approach is 
used; the approach choosing and matching system where the 
ideology of each approach, the associated tools, the inquiry 
system and the costs are examined.

3.2.2.4 NIMSAD

(Novich & Fitzgerald, 1988)

NIMSAD (Normative Information Model-based System Analysis and Design) proposes a framework which can be used to compare methodologies. This framework has eight stages:

1) Introduction to the real world - how methodologies depict and perceive the real world in which the problems exist.

2) Understanding the situation of concern, the concepts, models and theory - the ability of the analysts to gain a rich understanding of the problem, including the effects of their own values, models and relationships.

3) Diagnosis - the capture of the static representation of the problem (where are we now).

4) Prognosis outline - definition of expectations involving a critical evaluation of those expectations (where do we want to be and why).

5) Systems analysis - the conceptual mapping of the prognosis on the diagnosis, sometimes called the inter-gap analysis. This includes an analysis of why and what prevents achievement, and the identification of a relevant notional system.

6) Logical design - identifies the elements necessary to support the notional system.

57
7) Physical design - the analysis of the ways and means of achieving the logical design.

8) Implementation.

3.2.2.5 Avison & Fitzgerald

There are seven basic elements to the comparison model by Avison and Fitzgerald (1988):

1) Philosophy - the basis for the selection of area covered by the methodology, the systems, data or people orientation, the bias towards computerization and other aspects. There are four factors that need to be considered:

Paradigm - a specific way of thinking about problems encompassing a set of achievements which are acknowledged as the foundation of further practice. Two such paradigms are relevant here, the science paradigm and the systems paradigm. The science paradigm includes most of the hard scientific developments of this century. It breaks things down into smaller and smaller parts for examination and explanation. The systems paradigm takes on the holistic approach, concerning the whole picture, the emergent properties, and the inter-relationship between parts of the world.

Objectives - different methodologies have different objectives. Some concentrate on the development of a computerized information system. Others concentrate on the wider views of manual or organizational changes. These objectives will determine the boundary within
which the methodology will operate.

Domain - this refers to the area the methodology addresses and is closely related to the objectives. The development of solutions to different problems at different stages may be well co-ordinated. But it is often found that systems and their problems are inter-related, and the solution to a number of inter-related problems is different to the sum of the solutions to the individual problems in isolation. Therefore Avison and Fitzgerald suggested that in order to solve individual problems, it is necessary to analyze the organization as a whole and come up with an overall information system strategy. A top-down analysis of the organization is needed to ascertain the strategic requirements of the business, so that the right system is designed to meet these requirements.

Target - this refers to the particular types of problems, environment, or type or size of organization the methodology is specifically aimed at.

2) Model - this is the basis by which the methodology views reality. It is also a means of communication, capturing the essence of a problem and design, and provides insight into the problem or area of concern. There are four types of models: verbal; analytic or mathematical; iconic, pictorial or schematic; simulation. There are also two levels of describing the models. The logical level of description is one without any reference to the technology that can implement it. It provides the requirements specification of the system. The physical level of description includes the
technical information on particular parts of system implementation.

3) Techniques and tools - There are two types of technique: data analysis and process analysis. Sometimes techniques differ considerably from different methodologies, but the principle of the techniques is common and it is the notations and conventions that differ. Therefore it is good practice to identify the underlying principles of the techniques. Most of the techniques also bring up issues or concepts of importance in the first levels of operation before going into more detail. Therefore it is important that the technique guide its user through the process. The tools used are usually related to the methodologies themselves, or to the system development environment.

4) Scope - this refers to the stages of the life cycle of systems development covered by the methodology.

5) Outputs - this refers to the type, the quantity and the usability of the products of the methodology.

6) Practice - This is measured according to the methodology background, the user base, the participants in the methodology and the level of skill required.

7) Product - this is what the taker of the methodology actually gets for the money paid: software, documentation, types and amount of training agreed, advice service.
3.2.3 Summary

So far this chapter discusses the problems of comparing methods, which concern mainly the manner by which comparisons are carried out. The various ways to compare methods were then discussed. Feature analysis compares the methods against a list of features which has been used by others previously, or those newly created by the users. Another way is to compare general areas of methods such as their theoretical approach to systems design, and how they cater for cognitive aspects like human-computer interaction. A model, such as the one suggested by Avison and Fitzgerald, can also be used to give a comprehensive review of all aspects of each method.

One way to compare methods is to put each of them through a case. Such an investigation was carried out at Berlin and this will be discussed in the next section.

3.3 A Comparative Study on Methodology

A major investigation was carried out by the Technical University of Berlin (Olle et al 1986) between 1982 and 1985 on five system development methods:

- Structured Analysis and Design Technique (Ross 1977);
- Structured Analysis/Structured Design (De Marco 1978);
- Jackson System Development (Jackson 1983);
- Information System Analysis and Construction (Lundeberg, Goldkuh, Nilsson 1978);
- IBM-Verfahrenstechnik (the German version of IBM Improved Programming Techniques).
These methods were selected according to the general availability of published literature, relevance and availability for practical systems work in West Germany, and suitability for interactive application systems. The methods were introduced to students, tried out with a case study, and then evaluated. The case study used was a library system. A great deal has been learnt from the study concerning the particular methods and approaches to method characterization in general. However for the purpose of this research programme, the main focus is on the general criticisms emerging from the study across all the methods investigated. There are nine main points of criticism:

1) No one method deals with the issues of objectives, requirements, system functions, software design components and program components wholly, in a structured way, or helps to link them in a co-ordinated manner.

2) Selecting from the choice of concepts for modelling, their definition and meaning, and their applicability to what needs to be modelled is difficult.

3) The methods do not provide sufficient mechanisms for structuring, especially for developing a large system.

4) Only two methods studied deal with the requirements issues in any detail.

5) None of the methods deal with the area of man-machine interfaces satisfactorily, leading to poorly designed dialogue systems.

6) Only one method deals with parallelism of any kind.
7) A great deal of effort is required in all the methods to produce the necessary documents deriving from the development process. Even more effort is required if changes are required.

8) Although provisions have been made by all the methods to structure time processes, these often add to the number of problems encountered rather than solving any.

9) All the methods are essentially based on a linear view of systems development. New methods will be needed for an evolutionary approach based on development cycles.

The study concluded that the methods investigated were suitable for a medium-size systems development, with little human computer interaction and the functionality of the system clear in advance. The study stressed that the methods were not useless at all. But it should not be expected that one method would solve all systems development problems, nor would the shortcomings highlighted be eliminated by computer support of the methods. Finally the study stated that the methods should not be seen as static rule-based systems, but as processes of method development, use and revision based on experience. In doing so, the methods could become more dynamic and evolutionary, thus more of the real problems with systems development could be identified and solved.
3.4 Social and Human Factors Techniques Contribution

The problems highlighted by Olle et al and others are generally those of the social and human factors nature such as requirements elicitation and man-machine interface. Incorporating social or human factors techniques into the technical methods, will help to resolve such problems.

The objectives and requirements of the new system are important parts of the analysis and design process. Questions should be asked to find out the purpose of the new system or modifying an existing one before any work is carried out (Mumford 1986). Amongst them are:

- What are the reasons for building/modifying the system?
- What is the new/modified system required to achieve?
- Who has requested the system to be built/changed?
- What are the perceived benefits gained from it?

Without clear and unambiguous answers, it will be difficult to build a usable system which fits the needs of the organization (Checkland 1981). The answers are usually originated from the management and staff of the organization who are the ultimate end users of the system. These ideas reside in the minds of people and are not often explicitly expressed in any form. Most of the technical methods studied by Olle et al do not provide procedures or guidelines to elicit them from users. If techniques used in ETHICS (Mumford 1986) and SSM (Soft System Methodology, Checkland 1981) can be incorporated in the technical methods, then the end system should meet user objectives and requirements. In ETHICS, extensive questioning is carried out throughout the process to obtain the key objectives and the needs of the staff from the system. Some of the process concerning the
underlying objectives and requirements can be included in the
early part of a technical method, after which the rest of the
method can continue its route in producing the system. In
SSM, a lot of effort is spent in getting the underlying
reasons for the proposed system and its essential
specifications. These are referred to as the "system
thinking" activities in SSM. These "system thinking"
activities can be carried out during the early part of the
technical method, and the remaining stages can proceed to
produce the final required product.

Another problem with the technical methods is the lack
of provisions for man-machine interface and dialogue design.
In the study by the Technical University of Berlin, none of
the methods deal with this area at all. The structure and
effectiveness of the man-machine interface, the communication
between user and system, help to determine the usability of
that system (Shneiderman 1987). Procedures can be included
into a method which deal with the dialogue design and the
specification of interfaces. An example of this can be found
in the DIADEM methodology. In the methodology, there is a
technique called Job Stream Charting which handles task
allocation and job specification (DIADEM 1987). The
technique fits into the rest of the methodology which is
based on a combination of SSADM and ETHICS. One of the
functions of the technique is to determine the system
boundary and the locations for interfaces. The charts
produced show the type of information flowing between the
system and the user, which assist in constructing appropriate
screen designs and channels of communication.

The conclusion of the Berlin study stated that there
was little human-computer interaction in the methods covered.
Including some human factors techniques, such as the ones
suggested above, into the methods should be able to resolve some of the problems highlighted.

3.5 Experience in Integrating Human Factors Principles

In 1985 HUSAT was commissioned by a large UK organization to incorporate human factors guidance into structured analysis and design methods used by the organization. Damodaran et al (1988) argue that for nearly two decades, information systems have failed to deliver the benefits expected by users. One of the reasons cited is the inappropriate involvement of users in the design process. She points out that although structured system analysis need users to provide information, sometimes a substantial amount, for their work, this is normally as far as user involvement reaches. Users are not involved in decision making on key design issues. This often results in failure in the final system to cater for human and organizational needs. Damodaran et al (1988) also argue that effective user involvement is not just a matter of treating users as information provider. They should actively participate in the design decisions which relate to them. HUSAT’s aim was to make use of the schematic plan for the activities and decisions provided by the structured analysis methods throughout the design cycle, incorporating human factors techniques at relevant parts of the cycle.

The terms of reference agreed were as follow:

1) Identification of worksteps within the methodology which need to take into account human factors consideration;

2) Documentation of the relevant reference material;
3) The material developed has to be suitable for use by staff not trained in human factors techniques, and usable with little initial training and a minimum of direct consultancy support.

4) Guidance is drafted for the representative of the end users who will carry out the human factors activities.

Damodaran et al (1988) highlight some of the achievements of the integration work. From the earliest stages of the design process, crucial decisions relating to the allocation of functions between people and technology were made. However, human factors perspective should include thorough analysis of the user situation and user needs throughout the project life cycle. The appropriate steps necessary were introduced in the relevant stages of the methods. Another achievement is to provide procedures to assess the likely impact of any design decision taken on the human and organization areas.

Overall, several new techniques have been added: user analysis, task allocation procedures, job design and work organization. Significant expansion of the concepts and procedures associated with the following areas has also been achieved: user support specification, prototyping and definition of current problems. The major deliverables from this research are the new worksteps and procedures crucial to the user-centred design of systems, expanded worksteps to ensure human factors in all relevant activities are taken into account, and reference material in key areas of human factors developed to guide the staff involved in the design of systems.

Damodaran et al (1988) concede that there are still problems that are not resolved: - 
1) The need to codify rich, complex human factors expertise into simple guidelines for use by staff with little or no training in the behavioral sciences.

2) The need to transfer human factors knowledge to end users and consultants from different professional backgrounds sufficiently to achieve the required integration of human factors material.

3) The resource allocation and scheduling implications of expanding the methodology to include human factors work i.e. new activities have been added to the methodology for which no resources have been allocated and no elapsed time allowed.

4) The need for training and direct support in human factors experienced by users of the methodology.

Damodaran et al (1988) reported that several projects have been using the methodology enhanced with human factors techniques. Experience so far indicates that the steps in the methodology are practicable and appropriate and results in deliverables which reflects human factors expertise. Although it has been necessary for human factors consultants to provide careful training and assistance. This results in some user representatives have a growing knowledge of human factors and the system design context. However, it will be more desirable to educate users in human factors principles, as well as the user management, technical consultants, equipment manufacturers and suppliers. Damodaran et al (1998) conclude that changes to design methodology are necessary to ensure information systems in the future meet human and organization needs. Furthermore, the required training, support, assessment and reward procedure must also
be developed so that human factors principles are fully accepted as an important part of system design.

3.6 Summary

This chapter first discussed the difficulties in comparing information system design methodologies. The reasons for carrying out a comparison range from a pure academic investigation, to a practical search for the right method to use. The vast array of different methodologies and their methods, and the constant evolutionary change of design methodologies make it difficult to find a common ground on which proper comparison can be made.

The chapter then illustrated several methods of comparison. Feature analysis uses a list of important features as a basis by which different methods are compared. Another way to compare is by investigating the methods in key areas. Such areas include the theory or philosophy which underlies the method, an analysis of the workings of methods by undertaking case studies, and cognitive analysis of how the methods carry out the analysis work. Alternatively, the emphasis can be placed on the context of the problem at hand, and an appropriate method is sought to suit the circumstances. Other ways of comparison involve the detailed break down of the methods into their constituent parts, for comparison, one with another.

This was followed by the comparative study performed by the Technical University of Berlin. From the criticisms of the Berlin study, some suggestions were made on using social techniques within the technical methods. It then highlighted the work carried out by Damodaran to incorporate human factors principles into methodology.
The previous Chapter investigates the development of methods, showing a trend towards the incorporation of human factors. However, the previous Chapter concentrates on the theoretical arguments and it is important to consider what empirical evidence there is for their inclusion. This Chapter deals with comparison of methods. The Berlin study in particular provides the empirical evidence for the inclusion, or absence, of social factors. However, the study does not provide any empirical evidence for the use of a social-technical method. The next step is to look for such evidence, which will be covered in the next Chapter when considering DIADEM and its uses.

The suggestions on applying social techniques in technical methods mentioned above only cover part of the system design. Apart from helping to define key objectives, requirements elicitation, and dialogue design, these techniques can also help to design jobs in terms of task allocation and job specification. Amongst the techniques used in the DIADEM methodology, which combines the SSADM and ETHICS methods to cover both the technical and the social aspects of systems design, is one called Job Stream Charting. This technique is specifically designed and developed to deal with job design and work organization issues. However, DIADEM was used on a large scale project with mixed success. In the next chapter the methodology, and in particular the Job Stream Charting technique, will be discussed, together with the problems encountered in its use and suggestions for improvement.
4. Task Allocation and Job Specification

4.1 Introduction

Earlier in Chapter 2, the social approach to system analysis and design was discussed, in which emphasis is placed on the organizational impact on the business as well as the requirements and aspirations of the staff and their jobs. Later in the Chapter, the DIADEM methodology was also discussed in which procedures are provided to integrate human factors techniques with structured ones, so that the resultant system will be more usable and acceptable to users. One of the human factors techniques in DIADEM, which was developed to enable job design issues to be represented and considered in the early stages of system design, is Job Stream Charting (JSC).

4.2 Job Stream Charting

JSC is a technique developed by the Human Science and Advanced Technology (HUSAT) Research Institute at Loughborough University. Its development was based on the idea that the design of systems has traditionally failed to properly deal with the task allocation and job design requirements of the eventual system end users, resulting in poor system implementation (Ip, Damodaran, Olphert, Maguire 1990).

As we have seen, many design methodologies are predominately data-driven and technically oriented. The social and organizational aspects of the system are inadequately covered. Traditional system analysis concentrates on areas that will be computerised. Ip et al
argue that the whole organization should be viewed as an open system and a complex socio-technical system. Many activities and decisions will remain untouched by computerization, and hence will be ignored by the narrowly defined analysis process. Areas of work such as customer contact, flexible or adaptable working practice, most of the problem-solving activities and negotiation will still be predominantly human activities. These will have to be compatible with the incoming technical system for it to be functionally effective. This compatibility can only be achieved by producing a design specification which covers a broad spectrum of system analysis.

The narrow focus of the conventional system analysis results in an insufficient and very restricted representation of user requirements. Any flexibility claimed will be bound by the facilities provided by such a limited specification. This will result in the loss of the human skills, and thus true flexibility and adaptability of the newly installed system. Ip et al (1990) point out that none of the commonly used structured system design methodologies explicitly address the job design requirements of end users, even though it is generally accepted as fact that the design of a system and the tasks and jobs the users do are closely connected. They further argue that the authors of methods in which "job design" is mentioned have not supported their claims by providing the relevant tools and techniques, training and skill requirements in the methods. Neither do they have any provisions in the project management structure to cope with job issues. The effectiveness and influence of the analyst in job design is also limited. The analyst invariably follows strictly the methodology used. He will be concerned with the deliverables defined by the method, and this does not require him to show awareness of the impact of the proposed changes. He is usually not required to follow up
the new system, after its implementation, to experience for himself its strengths and weaknesses.

Poor communication is another factor in the difficulties between end users and system designers. Three reasons were given for this (Ip et al 1990):

1) difference in backgrounds in terms of training and expertise.

2) the different areas of interest and concern of both groups. The designers concentrate on the processes and data-flows, whereas the users are more interested in the operational aspects of the system at the interface.

3) The lack of suitable tools for communication between designers and end users.

The fundamental problem is a mismatch between users needs and the proposed system from the beginning of the analysis process. To solve this problem, techniques need to be developed to analyze the user organization in a more generic way, taking into account all technical and social aspects of system analysis. To overcome this a fully functional socio-technical system is required which can achieve its full potential.

Ip et al (1990) list three main aims of the development of JSC:

1) Determining the man-machine boundary of the proposed system. This should be done early in the design process as the acceptability of any task allocation will be assessed on its impact on users and their jobs. JSC helps to visualise such impact and assists in the subsequent
discussion and decision making process.

2) Exploring different job design options by illustrating the various ways in which the tasks can be allocated. The charts can show the significance of each option for different users. They can make comments about the charts, which will be evaluated by users and the management in order to select the most appropriate options.

3) An aid for communication between users and the designers. The charts can be kept as references to the decisions and choices made.

Using JSC to represent tasks and their functions, it is argued, will show that there is a direct relationship between system design and job design. It focuses designers' attention on the work processes occurring in the organization and therefore the fact that computers should be designed to support and to help humans to carry out these processes. JSC also shows that the interface is an outcome of task allocation decisions and not that of computer system design. The interface is the enabling facility for work processes to flow smoothly between the major components of the socio-technical system.

The principle terms used within the context of JSC are defined here:

- **Tasks** - the component operations within a job.
- **Activities** - those elements of the task performed by people. They consist of a series of activity steps.
- **Jobs** - specific groups of tasks assigned to people.
- **Job design** - the selection and combination of the most appropriate JSC options to form the work of a particular grade or group of staff.

74
Job Stream Charts (JSCs) - the products of the technique.

4.2.1 How to Construct the Charts

The first step in developing JSCs is to construct Task Allocation Charts (TAC). These charts show the sequence of tasks actions and decisions carried out by both humans and computers working together. The amount of detail shown in the TAC should be limited but sufficient to make clear the tasks involved. The human parts of TAC will then be used in the next step for the development of Job Stream Charts, while the computer parts will be used in the definition of the technical system.

The purpose of constructing Job Stream Charts is to investigate in more detail the tasks to be performed by people. They can show the possible ways in which the tasks can be assigned to different individuals or between grades of staff. When this is carried out for the full range of work processes which take place in the user area, it should be possible to evaluate all of the jobs which may be assigned to individuals and to apply job design criteria to them.

4.3 JSC Usage in DIADEM

The Job Stream Charting technique was used in the following work steps in the DIADEM methodology:

- Survey Current System: from user analysis, the allocation of tasks in the current system can be represented.
- Define Outline Functional Requirements: Job Stream Charts as input to the creation of conversation summaries.
- Identify User Acceptance Criteria: first-pass Job Stream Charts are drawn to represent options for each process on
the DFD to explore and define user acceptance criteria.

- Perform Sensitivity Analysis: the results of the activity "Identity key performance and sensitive areas" may identify the need for alternative allocations.

- Define Detailed Functional Requirements: make relevant changes to the Job Stream Chart interfaces and prototypes plan in the light of changes identified in the review of the current system survey. Design the user-user and user-computer interfaces. The Job Stream Charts drawn in the definition of job design and work organization indicate what data items have to be passed across an interface.

- Define Detailed Inputs and Outputs: determine the layout of the system's forms, reports and screens. The Job Stream Charts from the definition of job design and work organization are input to this activity.

- Define Job Design and Work Organization: document job design and work organization options in the form of detailed Job Stream Charts. Use the first pass Job Stream Charts drawn in the user acceptance criteria step as input.

4.4 JSC and the Design Teams

JSC is used in relation to the documentation of the current system, job design, work organization and interface design. Consequently, members of the systems design team should consist of representatives from users, experts in DIADEM, system analysts and designers. All these people should have sufficient knowledge and understanding about JSC and its purpose in order to contribute in the usage of the technique. Hence adequate training and education is necessary as part of the overall training programme in DIADEM.
4.5 Application of JSC

4.5.1 Task Allocation

There are often a number of ways in which work can be organised and tasks allocated between human and computer to produce the same end result. It is essential that as many of these options as possible are examined for each process. In doing so, the underlying critical job design requirements for the system will emerge. If they are then incorporated in the final system produced, it should then be an efficient one which is also acceptable to the staff themselves. Both user and technical teams should submit options to be discussed so that both the social and technical objectives can be met. The important point is to identify in general terms the kind of tasks to be given to both the human and computer components within the system to meet the requirements and objectives identified in User Analysis. These tasks will eventually be assigned to particular staff grades (in Job Stream Charts).

4.5.2 Job Stream Charts

After the allocation of tasks between the main users and computers is agreed and documented in the Task Allocation Charts, the issue of assigning tasks to the appropriate staff grade in order to provide input for job design is explored and developed. A number of Job Stream Charts are drawn up for each of the user functions. In order to create whole jobs for the people within the overall system it is necessary to select an option for each work process which allows it to be performed efficiently and provides satisfying job components. It is also essential to group the individual job components into a more complete and global view of jobs which meet specified criteria for good job design.
In general terms, a "good and satisfactory" job should have the following points:

- provide opportunity for learning and problem solving within the individual's competence.
- be seen as leading towards some sort of desirable future.
- provide opportunity for development in ways that are relevant to the individual.
- enable people to contribute to decisions affecting their jobs and their objectives.
- ensure that the goals and other people's expectations are clear and provide a degree of challenge.
- provide adequate resources (training, information, equipment, materials, time).
- provide adequate support from contact with others.

It has to be stressed that the creation of good jobs for the JSC option is not easy. While individual components taken from each of the selected option many appear satisfactory, the whole job may be too demanding, too simple or lacking in variety. Similarly while the job components in relation to a particular project application may appear satisfactory, this may not be the case when other tasks from other applications are included. Another important point to note is that job criteria are different for different grades of staff. For example, flexibility of a junior office clerk's job is very different from that of a senior departmental manager. Therefore, the system designers and the decision makers must take a holistic view of job design.

4.5.3 Interface Design

In JSC, when tasks pass from person to computer, they are indicated in the charts to show the human-machine boundary. This human-computer interface must be carefully
designed in order to support the communication necessary. Job Stream Charts are key components in the dialogue design process, which also needs material from data flow diagrams. In particular, JSC can help to identify human factors objectives for dialogue specification and evaluation. In order to produce an acceptable interface to users, there are general guidelines suggested (DIADEM 1987) which will help with the development of some typical tasks:

Data entry - flexibility, user input, user support, shortcuts for expert users, appropriate feedback.

Data amendment - user support, rapid access where necessary.

Locating a record - flexibility, appropriate feedback.

Browsing screens - user support, accessing relevant screen quickly, appropriate feedback.

Menu option selection - flexibility, meaningful words for options, appropriate feedback.

Job Stream Charts, therefore, aid the decision making process concerning the human-computer interface. The focus is taken away from the inputs and outputs of the automated system as the starting point for the design of that interface. Instead, the two mediums in the work process that the interface is meant to facilitate, the user and the system, are the important areas for considerations. The decisions made will be supported by user analysis data, HCI standards, guidelines and other user acceptability criteria.

Job Stream Charts also aid the decision making process concerning the human-human interface as they determine the form of communication between two jobs or grades. These decisions also determine form and report design for communication between the human components which should be consistent with those delegated to the automated system.
4.6 Guidelines for JSC

When JSC was being developed, guidelines were drawn up which were intended to help with the use of the technique:

1) Make sure both the user representatives and the designers are involved in the construction of the charts, and encourage them to present their ideas on task allocation and job design.

2) Ask all participants to consider the full range of alternatives that are conceivable. Ideas should not be immediately rejected on the grounds of cost or breaking the traditions of the organization. Deeper examination of options which at first seem unrealistic is often rewarding.

3) Construct TAC and JSC quickly as working documents which can then be modified in response to new information, user requests, feedback from walkthroughs. Only the final agreed version needs to be clean and tidy for future reference.

4) When JSC are used to gain an overview of the tasks assigned to various grades of staff, a colour scheme will be useful for tracing jobs in different charts.

5) The least amount of detail to be shown on charts should be enough to show the difference between options. If differences are clearly shown, it will aid decision making in job and interface design. If there is too little detail, the task boxes may be decomposed into lower level charts which expand on the information contained at the higher level.
4.7 Problems with JSC Application

4.7.1 General

During the development of JSC, several dangers were highlighted (Ip et al 1990) as being ones to avoid during the construction of Task Allocation and Job Stream Charts, in order not to undermine the overall objectives of JSC.

One such objective is that the technique should involve the staff or their representatives at the start of the analysis and design stage, as they will eventually be the end users of the system. Therefore, the charts should not be developed in isolation from the staff.

Another problem concerns the charts themselves. They can become too complicated or complex for users to understand, even though they may have knowledge about JSC. There may be a tendency to try to compress as much information as possible into the charts to give a more complete picture of the system. This will inevitably produce charts that are confused and overwhelm the user.

Users of JSC should also be aware that work done in other projects using JSC can be valuable to their own. Gaining access or information from other projects can help to form ideas and aspirations with respect to the system under consideration. However, it must be pointed out that each system is unique to its particular environment and users. Therefore, accessing material from other projects should not take the form of straight forward copying.

JSC is developed to be a flexible technique. Therefore, task allocation for a particular process should not be carried out too rigidly. It is often necessary to
modify the charts during their construction in the light of changing ideas, goals and new information. So one must be prepared to receive, analyze and evaluate these new inputs, and decide how to incorporate them into the charts.

4.7.2 Using JSC

In the project where the DIADEM method (and hence JSC) was used, the project teams were divided into two main groups: the user team and the design team. Both teams were further subdivided into smaller teams working on different areas or modules of the project. The user teams dealt with issues such as job design, work organization, user acceptance criteria, education and training. They were primarily concerned with the end users and their requirements. The design teams dealt with the technical, logical, and physical design of the system as well as the design of the actual interfaces. This organization was specified in the DIADEM methodology and was broadly followed in the project. The responsibility for using JSC was placed upon the user teams. Team members were trained in the use of JSC, and then allocated to the design teams. There they assisted in the construction of Task Allocation and Job Stream Charts. Both user and design teams also assisted in the decision making stage when the social and technical aspects of the system took shape.

When JSC was used by the organization, problems arose with the application of JSC as well as JSC itself which, in the end, affected the effectiveness and the success of the technique.

The first problem that emerged soon after the start, was the apparent competition between both groups (however this is not a problem specific to DIADEM). The user group
had the benefit of human factors consultants. The design group received advice from a renowned international technical consultancy. The technical aspects tended to dominate the project. As a consequence, the work on the social and human aspects of the system did not acquire as much attention as was necessary. This bias need not necessarily be attributable to any one party. However, the fact that the technical advisors dominated project meetings, suggests that the project management tended to deal more with the "hard" issues of the technical specification and equipment, rather than the "soft" issues of job design and work organization (Olphert et al 1990).

The second problem was the production of the charts, or the apparent over-production of them. JSC does require its users to draw up charts for all the options that come up during job design as well as reflecting the current status of the jobs themselves. However, it seems that a lot of effort was spent in the project simply producing many of the charts, with no clear statements from the user groups as to why they had been produced or what would be achieved by constructing them (Olphert et al 1990). The charts however did cover many jobs and many options. They ranged from the mundane jobs of mail sorters and messengers to counter receptionists, supervisors and managers.

The third problem was the apparent inability to make use of the charts during discussion and decision making. During these activities, it was noted that whenever job design issues came up, the project managers would often ask members of the user teams how the work should be organised. They did not then apparently use the charts as reference or backup to what they had said. They tended to rely on their knowledge and experience. The teams did not have time to organise the charts to obtain the relevant information on job
design and work organization. Therefore, what the user team members said about work organization was based on their own experience. This being subjective, or even tailored to satisfy the management, did not necessarily reflect the desirable structure of the organization.

The fourth problem was to do with the technique itself. After discussions with a key member of staff from HUSAT who was involved in the project, some of the fundamental inadequacies of JSC as a technique for specifying job design requirements emerged.

Firstly, there was uncertainty as to what and how much information on job design needed to be considered in the system specification. This affected the usage of JSC in that the user group did not know how far it had to go and how much detail was required in a JSC.

Secondly, it was not clear to the system designers how to make use of the information from JSC. This could be due to lack of emphasis by the organization, or the management view of the importance of job design issues in systems development. Another possibility could be the lack of guidance in DIADEM itself. Therefore the JSC charts tended to be neglected or ignored during crucial decision making concerning the arrangement of jobs and staffing for the new system.

Thirdly, the information obtained from the JSC was not always sufficient to design jobs. In particular, the JSC did not provide information such as volume, timing, frequency, and dependency. It should also have provided other information such as the organizational objectives and priorities as well. This information is essential for complete job design.
To sum up, the four problems experienced by project members in using JSCs during the project were: the competition between job design and technical design, uncontrolled production of the charts, inability to make use of the JSC information and the deficiencies in the JSC as a representation.

4.8 Some Suggested Improvements

From the discussions with one of the human factors consultants involved in the project, together with the available documentation on JSC, such as project reports, and the large quantity of DFD diagrams and JSC charts, it became clear that there is a need to improve the use of JSC. There are several areas in which changes could be made. These include:

1) the inclusion of Task Allocation Charts (TACs) in the final products of JSC to provide an easier introduction to JSC in meetings and discussions,

2) a cross referencing system between the relevant sections of JSC charts and DFD diagrams to link up the two techniques,

3) the introduction of a task catalogue to detail the precise description of each job,

4) an improved training and education programme on job design principles and the JSC technique itself.

The suggested improvements will of course add to the workload and development cost of projects using the methodology. These will have to be taken into account when considering their merits.
4.8.1 Inclusion of TACs

The inclusion of TACs as the first level of the final JSC charts will enable the charts to become more usable. In the DIADEM project, users of JSC commented that the charts were time consuming to produce and difficult to make use of during later project work and discussions. The charts were presented in great detail, without leveling, and it proved difficult to make decisions on which option to take. However, the Task Allocation Charts, which were the basis for the construction of Job Stream Charts as well as fulfilling other functions earlier, were not used at all after the initial stage. If these charts became the first set of documents to be presented in discussion, the users would find it easier to understand the implications of different options, and thus be better informed and prepared for the more detailed Job Stream Charts which would become second level documents.

Figure 1 shows an example of a JSC representing the receiving and assessment of an insurance claim. This would be the Job Stream Chart representing an option on receiving and locating clients' claims. In the DIADEM project, this level of detail was the only one used. But in early project discussions, to identify potential options for the computer system architecture, the grade of staff involved and the method of retrieval are not particularly relevant. In Figure 2, a Task Allocation Chart is shown for the same process but at a higher level of representation. This is far simpler and would be adequate for the decision making needs of the project at the early stages, less time consuming to produce and would provide a reference point from which to draw subsequent Job Stream Charts.

By presenting this chart during discussion, the essential information of the human machine boundary can be
Figure 1 A job stream chart of the current system
Figure 2 A task allocation chart of the current system
clearly seen. Tasks done by human and computer are also clearly seen, with the necessary interface location marked. Together with the cross referencing system (which will be discussed in the next section), the inclusion of TACs should help in the discussion and decision making and together with other analysis and design documents representing the various business options for the proposed new system.

With the inclusion of TAC as a first or top level representation of JSC, the numbering system of all the charts will need to be changed in order to reflect the levelling of the charts. Such a system can be similar to that used in the DFD. For example, in the DFD system the different levels of diagrams will be numbered as follows:

- Top Level DFD
- Second Level DFD
- Third Level DFD

1  
1.1 
1.1.1

Similarly, JSC can also be numbered in the same way:

- Top level JSC
  (Task Allocation Chart)
- Second Level JSC
  (Job Stream Chart)

1  
1.1

A possible next step will be to split up the job stream chart, now as a second level representation, into several simpler charts. This will further assist the understanding and usage of the charts by reducing the amount of information contained in each chart. Each one of these charts can represent a job grade, showing its responsibilities and where it interfaces with the computer system. These charts will be numbered to relate them to the original TAC and the first job stream chart. Figures 3 and 4 show the separate representation of the Junior Clerk and the Claims Clerk. The
<table>
<thead>
<tr>
<th>JSC 1.1.1</th>
<th>Receive Claims Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Junior Clerk</strong></td>
<td><strong>Computer</strong></td>
</tr>
<tr>
<td>Opens post</td>
<td></td>
</tr>
<tr>
<td>Sorts claims</td>
<td></td>
</tr>
<tr>
<td>Seeks claim information</td>
<td>Retrieve information</td>
</tr>
<tr>
<td>Receives information</td>
<td></td>
</tr>
<tr>
<td>Sends information to Claims Clerk</td>
<td></td>
</tr>
<tr>
<td>Notifies claimants</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3 The current job stream chart of the Junior Clerk
Figure 4  The current job stream chart of the Claims Clerk
two charts come from the original job stream chart, but since they are now separated, more detail can be shown on each chart for the respective job grade.

There are problems with splitting the job stream chart. Some duties and responsibilities are straightforward and easy to assign to a particular job grade. However there are occasions where several members of staff may have to carry out the same duties, regardless of rank or grade. Therefore, creating the charts by job grades will result in several of the jobs appearing on different charts. This becomes important since during discussions all the relevant charts of the area concerned must be presented.

Another potential problem arising by splitting up the original job stream chart by job grade is that user team members may take this opportunity to raise issues on the redistribution of duties and responsibilities. Although one of the functions of JSC is to analyze the task allocation of the system, this can also create friction between management and staff about reallocation of existing tasks and new additions to the new system. This can only be resolved by discussions within key personnel when the situation arises.

4.8.2 Cross Referencing

A crossing referencing system for the charts and diagrams of JSC and DFD would be useful to keep track of documentation from both techniques for discussion purposes later on. Since both the design teams and the user teams are trying to design the proposed system according to the specification, a system to link the relevant sections of common areas together will help the decision maker to understand how processes and jobs are related to each other in all areas, and how any proposed changes to either aspect
of the system will affect the other. However, during the DIADEM project there were virtually no such references between documents. Discussion with the HUSAT personnel involved in the projects led to the conclusion that a cross referencing system would be beneficial to the overall system development. The following two reasons were cited.

Cross references between DFD diagrams and JSC charts dealing with the same area of the organization will enable the technical consultant to keep track of the social consultant's work and ideas about the area concerned, and vice versa. This will help with the accurate development of the charts and diagrams because it also helps both the user and design teams to appreciate the importance of both techniques in the design stages. When JSC was used in the project, it was noticed that both JSC and DFD were being used and developed quite independently without reference to each other. Subsequently, one team did not know what the other team had done. Some of the diagrams and charts in the same area did not correlate with each other.

The cross referencing process will be carried out after both sets of documents have been prepared. Leaders from the design teams and users teams involved in the production of both the JSC charts and the DFD diagrams will meet to produce the cross references. The technical and social consultants should also be present at the beginning in an advisory role. This may well be the first time either side comes across the charts and diagrams. The consultants can explain the symbols, notations and meaning of their respective documents. At the first meeting, the need for cross referencing will be explained to the group. Then a system should be agreed by which referencing will be done. The system may involve some combination of each set of document's own identification system, or it can be a totally new one to distinguish it from
the identification systems of both sets of documents. When such a system is agreed, the group can then go through both sets of documents and identify common areas represented by the charts and diagrams. They will be marked with the agreed referencing system to indicate that in future, when these areas come up for discussion, both sets of documents can be retrieved.

One of the purposes of cross referencing is to enable both user teams and design teams to identify common areas of their respective work. Each set of JSC charts and DFD diagrams are produced with their own procedure and technique. Cross-referring between the two should not affect the overall outcome of the analysis work. Both techniques represent different aspects (jobs and processes) of the same area (of the organization) or a business option. Therefore, one technique should not influence the workings of the other and vice versa. However, because of how cross references are made, each side has the opportunity to see and comment on the work done by the other. It is possible that the design team may indicate that some of the user team's design options will be difficult to implement within the current technical specification or feasibility. The user team may also point out that parts of the technical design will be difficult to fit into the tasks and jobs in the organization. These potential problems can be tackled here since both teams are together, and will not be left until well into the development stage when it will be too late to resolve them. If there are changes to the specification, the cross referencing system should enable both the relevant JSC charts and DFD diagrams to be identified and changed. Any implication of the technical and social changes affecting each other can be discussed at the first opportunity in these cross referencing meetings.
With the different levels of the DFDs and JSCs (discussed in the previous section) involved, the referencing must be clear so that the correct diagram or chart is referred to. When a common area is identified, the top or first level of the relevant set of DFDs and JSCs are marked and cross referenced first. Afterwards, the second levels of the charts and diagrams will then be examined and marked according to the agreed system. Ultimately, all relevant and compatible areas (processes in the DFDs and the job descriptions in JSCs) will be cross referenced. The lower level cross referencing must be done to all levels so that when changes are made to the proposed system, all the relevant sections of both sets of document are modified accordingly.

All cross references made are entered into a table to keep a record of the process. The cross referencing table will contain three parts. The first part is the function area of the system which the charts and diagrams represent. The next part will be the JSC chart number and its title. This should identify the particular chart, but document number and page number can also be included for ease of location. The third part is the DFD diagram number and its title. This identifies the DFD diagram concerned with additional document number and page number for easy access.

Figures 5, 6 and 7 are examples of cross referencing between top level job stream charts and data flow diagrams in a related area and the referencing table. This example shows the assessment part of an insurance process upon receipt of a claim. It must be pointed out that this example has been invented for the purpose of illustration.
Figure 5 The top level current data flow diagram of the system with JSC references.
Figure 6: The current job stream chart of the Junior Clerk with DFD references.
<table>
<thead>
<tr>
<th>JSC 1.1.2</th>
<th>Receive Claims Applications</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Claims Clerk</strong></td>
<td><strong>Computer</strong></td>
</tr>
<tr>
<td>Receives information from Junior Clerk</td>
<td></td>
</tr>
<tr>
<td>Assesses claim DFD 1.4</td>
<td></td>
</tr>
<tr>
<td>Fixes claim DFD 1.4</td>
<td></td>
</tr>
<tr>
<td>Updates claim file</td>
<td></td>
</tr>
<tr>
<td>Sends claim decisions to Junior Clerk</td>
<td></td>
</tr>
</tbody>
</table>

Figure 7 The current job stream chart of the Claims Clerk with DFD references
A table is also constructed to show the related areas:

<table>
<thead>
<tr>
<th>Function/Area</th>
<th>JSCs</th>
<th>DFDs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Claim Assessment</td>
<td>1.1.1</td>
<td>1.1</td>
</tr>
<tr>
<td>Junior Clerk</td>
<td></td>
<td>Receive claim</td>
</tr>
<tr>
<td></td>
<td>1.1.2</td>
<td>1.2</td>
</tr>
<tr>
<td></td>
<td>Claims Clerk</td>
<td>Sort Claim</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Notify</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Assess Claim</td>
</tr>
</tbody>
</table>

4.8.3 Task Catalog

The idea behind the task catalogue is to assist the users to formulate ideas about a particular job. These ideas include listing out the main purpose of the job, what grade of staff can take on the job, whether any authorization or supervision is necessary, where the job fits in the organization's working practices, any inter-dependencies between this job and others, the frequency that the job has to be performed, and the volume of the work anticipated. This catalogue should be compiled after the initial Task Allocation Charts are completed. In doing so, the users will have a clear idea of what is required in the job and can subsequently construct Job Stream Chart options. It is also hoped that the catalogue can be used as a reference should there be enquiries or confusion when the charts are studied.

Using the example of the insurance company, the following is an illustration of the task catalog of a junior clerk:
Department
Claims and Assessments.

Position in the Department.
Junior Clerk.

Job Title in the System
Junior Clerk.

Job Description
1) Receive claims by post or by interviewing client at office.
2) Sort claims into categories.
3) If new account, enter details into new file. Otherwise, retrieve client information from file.
4) Send claims to Claims Clerk of relevant sections.
5) Receive decisions from Claims Clerk.
6) Inform Client of decision by post, update client file.

Authorization and Supervision
No specific authorization required.
General supervision from senior clerk.

Key Personnel Connected with Job
Claims Clerks from all sections.

Frequency and Volume of Work
Average workload.
Sudden increase after bad weather or disaster.

4.8.4 Training

A different approach to training and education in job design and the use of JSC came about during discussion with the HUSAT staff involved in the DIadem project. It should be pointed out here that the author does not have the necessary qualifications or experiences to suggest a comprehensive training programme. However, there are a few points that are worth considering. One is that, at the beginning of the
training programme, the concepts and relevance of job design and work organization can be put across in a manner which will emphasise their importance to the effectiveness of the commercial system design process. However it must be stressed that this should not be done in too great a depth as this may alienate the users. Another point is that during the DIADEM project, the social consultants were acting in the advisory role of indicating what needed to be done in JSC and on what basis to evaluate job options. But they were not in a position to recommend to the project the "best" way of doing any given job, nor did they actually become involved in the work. The result was that frequently the JSC users expressed uncertainty about the method and the objectives during the analysis and development work. In other circumstances it would be desirable to involve the skills and expertise of the social consultants more in the use of JSC without compromising their advisory role, i.e. from advising people how to use JSC to actually doing the JSC work for them. Since each project and its project teams are different, the degree of involvement can only be established during the initial training period.

A suggestion on training is to use the user teams themselves as a case study. JSC can be used to analyze the jobs involved in a system design project team. This can make the training programme more appropriate and interesting since they are using the technique on themselves. But it must be pointed out that this is only a case study during training, and need not be taken too seriously as this might create friction and problems within the user teams before work starts.
4.8.5 An Example

This example is based on the Postgraduate Research Admissions System of the Department of Computer Studies. In the example, the main purpose of the analysis is to transfer some of the functions of the Postgraduate Research Admissions system onto computer, in order to fit in with the computerization of the University Postgraduate Office. All information concerning applicants would be sent on line via an inter-departmental network. The data flow diagram is based on work carried out by students who have been analyzing the Admissions System using SSADM. Information for the job stream charts was obtained from interviews with the Admissions Tutor and the Admissions Secretary.

Below is the task catalogue of Postgraduate Research Admissions Tutor:

Department
School of Pure and Applied Science, Department of Computer Studies.

Position in the Department
Senior Lecturer.

Job Title in the System
Postgraduate Research Admissions Tutor.

Job Description
1) Receives applications from Postgraduate Research Admissions Secretary.
2) Examines the research topic in the application.
3) Examines applicant's qualifications.
4) Examines applicant's funding arrangements for the proposed research.
5) Checks research topic with a list of potential Research Supervisors and their areas of interest.
6) Circulates copies of applications to possible Research Supervisors.
7) If there is no positive response, instructs Postgraduate Research Admissions Secretary to inform the Postgraduate Office of the University's Administration of the Department's rejection.
8) If there is a positive response, instructs Postgraduate Research Admissions Secretary to inform the Postgraduate Office of the University's Administration of the Department's offer of a place.
9) Allocates accepted applicant to a Director of Research.
10) Applies for grant for suitable home students from the Scientific and Engineering Research Council (SERC).

Authorization and Supervision

No need to be authorized or supervised. The Postgraduate Research Admissions Tutor is the sole decision maker.

Key Personnel Connected with the Job

Postgraduate Research Admissions Secretary.

Frequency and Volume of Work

Application received through the year, but generally increase from January until May.

Other Relevant Points

1) Replies to direct enquiries.
2) Updates the list of potential Research Supervisors and their areas of interests.
3) Redirects applications back to the Postgraduate
Office to be forwarded to other more appropriate Departments.

Below is the task catalogue of Postgraduate Research Admissions Secretary:

Department
School of Pure and Applied Science, Department of Computer Studies.

Position in the Department
Secretary to the Professor of Computing in the Department.

Job Title in the System
Postgraduate Research Admissions Secretary.

Job Description
1) Receives applications from Postgraduate Office of the University Administration.
2) Photocopies application forms. Keeps originals in files and sends copies to the Postgraduate Research Admissions Tutor.
3) Attaches special forms to the circulating document to possible Research Supervisors.
4) Notifies Postgraduate Office of the Department's decisions on applicants.
5) Sends forms to Research Supervisor and Director of Research of the accepted applicant to sign.
6) Sends letters to referees listed by the accepted applicant for references.
7) Sends lists of accepted applicants to the Administrative Assistance of the Department for the allocation of office space.
Authorization and Supervision

None.

Key Personnel Connected with the Job

Postgraduate Research Admissions Tutor, Research Supervisors, Departmental Administrative Assistance, Postgraduate Office Personnel.

Frequency and Volume of Work

Application received through the year, but generally increases from January until May.

Other Relevant Points

1) Replies to direct enquiries.
2) Sends grant application to the Scientific and Engineering Research Council (SERC).

Figure 8 shows the top level Current Physical Data Flow Diagram (DFD 2) of the Admissions System. All applications are sent from the University's Postgraduate Office to the Department's own Postgraduate Research (PR) Administration (Process Box 2.1). This is the central part of the Admissions System, with the Admissions Secretary as the key personnel. The applications are filed and passed on to the Admissions Tutor to be assessed (Process Box 2.2). Apart from examining an applicant's qualifications and funding arrangement, the Admissions Tutor has to look for a suitable supervisor who is interested in the proposed area of research. After the Admissions Tutor has accepted the applicant, he allocates a Director of Research to him before sending the decision back to the Secretary who then seeks formal approval from the Head of Department (Process Box 2.3). Afterwards, applicant files are updated before informing the Postgraduate Office of the Department's decisions. All related administrative work concerning the
Postgraduate Research Admissions System

2.1 PR Administration
Handles applications and enquiries, communicates with Postgraduate Office

2.2 Assessment
Evaluates research proposals, finds Supervisors and allocates Director of Research

2.3 Head of Department
Gives approvals to recommended applicants

2.4 Departmental Administration
Allocates office space for accepted applicants

Figure B The top level Current Physical Data Flow Diagram of the Admissions System
Figure 9: The second level Current Physical Data Flow Diagram showing P R Administration.
Figure 10 The second level Current Physical Data Flow Diagram showing the Assessment process.
Department is also initiated. The Postgraduate Research Administration also handles direct enquiries from prospective applicants, as well as contacting referees of accepted applicants for references.

Figure 9 is the second level Current Physical Data Flow Diagram of the P R Administration (DFD 2.1). This shows in more detail the processes carried out by the Administration, where it receives information from, and where it sends information to. Figure 10 is the second level Current Physical Data Flow Diagram of the Assessment (DFD 2.2). This shows in more detail the processes involved in assessing applications.

Figure 11 is the Current Task Allocation Chart (JSC 2), the first level Job Stream Chart showing the Admissions System and the jobs carried out by humans and the computer. Within the Admissions System, humans handle the receiving of applications, their assessment, and the decisions made on them. The computer handles information about applicants, records those who have been accepted, and holds standard letters of reply to enquiries and request for references. Figure 12 is the second Current Job Stream Chart (JSC 2.1) showing in more detail the jobs and responsibilities of the Admissions Secretary. Figure 13 is the second level Current Job Stream Chart of the Admissions Tutor (JSC 2.2).
Figure 11 The Current Task Allocation Chart, the first level Job Stream Chart of the Admissions System
Figure 11 The Current Task Allocation Chart, the first level Job Stream Chart of the Admissions System
<table>
<thead>
<tr>
<th>Admissions Tutor</th>
<th>Computer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DFD 2.2.1</strong> Receives applicant information</td>
<td></td>
</tr>
<tr>
<td><strong>DFD 2.2.2</strong> Examines applicant qualifications</td>
<td></td>
</tr>
<tr>
<td><strong>DFD 2.2.3</strong> Examines funding arrangement</td>
<td></td>
</tr>
<tr>
<td><strong>DFD 2.2.4</strong> Finds supervisor</td>
<td></td>
</tr>
<tr>
<td><strong>DFD 2.2.1</strong> Sends decisions back to P R Secretary</td>
<td></td>
</tr>
<tr>
<td>Applies grants for home students</td>
<td></td>
</tr>
</tbody>
</table>

Figure 13 The second level Current Job Stream Chart of the Admissions Tutor
The following is the cross referencing table for the current data flow diagrams and job stream charts:

<table>
<thead>
<tr>
<th>Area</th>
<th>Data Flow Diagram</th>
<th>Job Stream Chart</th>
</tr>
</thead>
<tbody>
<tr>
<td>Admissions System</td>
<td>DFD 2</td>
<td>JSC 2</td>
</tr>
<tr>
<td>P R Administration</td>
<td>DFD 2.1</td>
<td>JSC 2.1</td>
</tr>
<tr>
<td>Assessment</td>
<td>DFD 2.2</td>
<td>JSC 2.2</td>
</tr>
</tbody>
</table>

An interesting point came up during the cross referencing process. There was initial uncertainty as to which job stream chart should be referred to in the data flow diagram. In the end, the first level job stream chart, the task allocation chart, and the top level data flow diagram were cross-referred. The second level data flow diagrams were cross-referred with the second level job stream charts. In larger and more complex system analysis, care should be taken to correctly refer to the relevant charts and diagrams during referencing.

It is important at this point to be clear at what stage the analysis process has reached. Figure 14 is the standard SSADM structure showing the six stages of SSADM. A more detail description of SSADM can be found in Appendix A. The charts and diagrams discussed so far would constitute part of the end products from Stage One. The next stage is to define the requirements of the system. Figure 15 shows the top level Current Logical Data Flow Diagram of the Admissions System (DFD 3). It should be pointed out here that SSADM has its own diagram numbering system when moving from one stage to another. But in this example, each new set of diagrams is simply numbered sequentially for illustration purposes. It
Figure 14 The stages in an SSADM project
Figure 15  The top level current logical data flow diagram of the Admissions System
was decided that the current logical system was adequate enough to cope with the computerization. Therefore, it would not be changed. The requirements then were to find ways to integrate with the computerization of the Postgraduate Office, keeping the logical structure of the current setup. The next stage would be to select an appropriate business system option to be developed.

Two of the business system options will be used in this example. Option one will have the applicant's information received directly from the University Postgraduate Office's system into the Department's system via an inter-departmental computer network. This information will stay within the Department's system. It is sent to the Admissions Tutor via an internal communication network. His decision will then be sent back to the Postgraduate Office's system. The main emphasis here is to have the applicant's information stored within the systems and transferred via networks in order to reduce the paperwork generated in the present setup and to fit in with the computerization of the Postgraduate Office. Option two will have the applicant's information sent directly to the Admissions Tutor from the University Postgraduate Office's system via an inter-departmental network. The Tutor is the only person in the Admissions System who decides the success or failure of application. His decisions will be sent back directly to the Postgraduate Office. The Admissions Secretary will then receive information of accepted candidates, and start the next steps of the Admissions procedure: confirmation with supervisors and Directors of Research, requests for references. The main emphasis here is also to fit in with the computerization of the Postgraduate Office. It also takes the opportunity to reorganise and reallocate some of the responsibilities between the Admissions Secretary and Tutor.
The design team may favour option one as this will not alter much the current setup of the Admissions System, and will offer plenty of opportunity to develop the hardware and software necessary to receive and store information from the Postgraduate Office. The design team will also investigate whether existing network within the Department can be utilized to carry information to the Admissions Tutor, or whether a new network will have to be installed. The effect option one will have on the human part of the system can be seen through the job stream charts JSC 4, JSC 4.1 AND JSC 4.2. The following is the job description part the task catalogue of the Admissions Secretary under option one:

**Job Description**

1) Receives applications from Postgraduate Office of the University Administration.
2) Sends applications to the Postgraduate Research Admissions Tutor.
3) Circulates applications to possible Research Supervisors.
4) Notifies Postgraduate Office of the Department's decisions on applicants.
5) Sends forms to Research Supervisor and Director of Research of the accepted applicant to sign.
6) Sends letters to referees listed by the accepted applicant for references.
7) Sends information of accepted applicants to the Administrative Assistance of the Department for the allocation of office space.

Figure 16 shows the task allocation chart (JSC 4) of the Admissions under business system option one. The major difference between this chart and the task allocation chart of the current system is the use of the computer to receive applications in the first place before they start to go
Figure 16 The Task Allocation Chart of the Admissions System under business system option one
Figure 17 The second level Job Stream Chart of the Admissions Secretary under business system option one
Figure 1B The second level Job Stream Chart of the Admissions Tutor under business system option one.
through the system. Figure 17 shows the job stream chart (JSC 4.1) of the Admissions Secretary under business system option one. It shows that the Secretary will now have more contact with and usage of the computer, compared with the existing working practice, in order to carry out her Admissions work. Figure 18 shows the job stream chart (JSC 4.2) of the Admissions Tutor under business option one. This shows that the Tutor will now receive application information from the computer, and send his decisions back via the computer. Both the Secretary and Tutor may or may not agree to these changes. Their reactions constitute a major part of the impact option one has on the Admissions System. To assess such an impact means gathering reactions from personnel from the technical and social sides of the system. And to ensure both parties's views are adequately represented and brought up during discussion, documents of the relevant areas need to be presented together and cross-refer to aid discussion. With option one, it will be the data flow diagram DFD 3 and the job stream charts JSC 4.1 and 4.2. It may also be necessary to bring in the charts and diagrams of the current system to assess the true impact option one has.

Option two involves a more substantial change from the existing system. The Admissions Tutor now directly receives the applicant's information on line from the University's system. He will then send his decision back directly to the University, before informing the Admissions Secretary to carry out the necessary administrative work. Again in this option, it will involve the development of the hardware and software needed for communication between the Postgraduate Office and the Department, as well as the Department's own Admissions setup. The following is the job description part of the task catalogue of the Admissions Secretary under option two:
Job Description

1) Receives information of accepted applicants from the Admissions Tutor.
2) Sends forms to Research Supervisor and Director of Research of the accepted applicant to sign.
3) Sends letters to referees listed by the accepted applicant for references.
4) Sends lists of accepted applicants to the Administrative Assistance of the Department for the allocation of office space.

Figure 20 shows the job stream chart (JSC 5.1) of the Admissions Secretary under option two. This shows that the Secretary will now carry out fewer duties. However, the Admissions Tutor's duties have increased under option two.

The following is the job description part of the task catalogue of the Admissions Tutor under option two:

Job Description

1) Receives applications from Postgraduate Office of the University Administration.
2) Circulates applications to possible Research Supervisors.
3) Notifies Postgraduate Office of the Department's decisions on applicants.
4) Sends information of accepted applicants to the Admissions Secretary.
5) Applies for grant for suitable home students from the Scientific and Engineering Research Council (SERC).

Figure 19 shows the task allocation chart (JSC 5) of the Admissions System under business system option two. This is the same chart as the one under business option one as the
Figure 19 The Task Allocation Chart of the Admissions System under business system option two
Figure 20 The second level Job Stream Chart of the Admissions Secretary under business system option two
Figure 21 The second level Job Stream Chart of the Admissions Tutor under business system option two
tasks carried by humans and computer are the same, as well as the interfaces between the two. Figure 20 shows the job stream chart (JSC 5.1) of the Admissions Secretary under business system option two. This shows that the Secretary now has fewer tasks and duties to perform. Figure 21 shows the job stream chart (JSC 5.2) of the Admissions Tutor under business system option two. This shows that he will have a more substantial contact with and usage of the computer part of the system, and is himself responsible for receiving the applicant's information and the notification of his decision. Both the Admissions Tutor and Secretary have other responsibilities in the Department away from the Admissions System. Therefore, the impact option two has on them will be enormous and must be taken into consideration. So the relevant charts and diagrams can be cross-referred and brought up in support during the discussion of the option. Again it may be necessary to bring in charts and diagrams of the current system to highlight the impact this option has.

The following is the cross referencing table for the data flow diagrams and job stream charts under the two options:

<table>
<thead>
<tr>
<th>Area</th>
<th>Data Flow Diagram</th>
<th>Job Stream Chart</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Option One</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Admissions System</td>
<td>DFD 3</td>
<td>JSC 4</td>
</tr>
<tr>
<td>PR Administration</td>
<td>DFD 3</td>
<td>JSC 4.1</td>
</tr>
<tr>
<td>Assessment</td>
<td>DFD 3</td>
<td>JSC 4.2</td>
</tr>
</tbody>
</table>

126
In this section, some of the suggested improvements in the previous section were illustrated by using the example of the computerization of the Postgraduate Research Admissions System of the Department. In particular, the example showed what a task catalogue is, how the task allocation charts could be presented as first level job stream charts, and how cross referencing could be used to ensure that the impact on the technical and social aspects of the system are properly considered. The example showed how the current job stream charts were produced along with the current physical data flow diagrams, and how they could be cross-referenced. After the logicalisation process, job stream charts were also produced for each business system option to highlight the impact each one would have. In this example, the current logical processes were kept unchanged, and the business system options were constructed to integrate the department's system with that of the Postgraduate Office. In other cases, the corresponding logical data flow diagrams for each business system option would be produced as well. With option one, its main impact would be an increased computer workload on the Admissions Secretary. This could be seen through job stream chart JSC 4.1 (Figure 17). With option two, its main impact was the transfer of some of the processes from the administration to the assessment part of the system. This also means that some of the Admissions Secretary's responsibilities are also transferred to the Admissions Tutor. This could be seen through the job stream charts JSC
5.1 (Figure 20) and JSC 5.2 (Figure 21). The relevant charts and diagrams should be used as supporting evidence during discussion so that each option can be properly assessed. Cross referencing the charts and diagrams will ensure that the correct material is being used in this discussion.

4.9 Summary

In this Chapter, the technique of Job Stream Charting was examined. Its development, application in a project and the problems encountered were discussed. They were identified as follows: a lack of coordination and communication between the social and technical design teams regarding their respective work; difficulties in interpreting and making use of the charts from Job Stream Charting. Four possible improvements were suggested with illustrations to three of them to demonstrate their feasibility:

1) the use of Task Allocation Charts (TACs) as the first level of JSC, and the job stream charts as second level representations;

2) a cross referencing system between JSC charts and DFD diagrams;

3) a task catalogue to detail the precise description of each job;

4) a training and education programme on job design principles and the technique itself.

These suggested improvements will help to resolve the four problems identified in the earlier section: the competition between job design and technical design,
uncontrolled production of the charts, inability to make use of the JSC information and the deficiencies in the JSC as a representation. Using TACs as first level representation of JSC serves two purposes. One is to make the products of JSC less complicated to construct and understand. The levelled and split charts should help with the discussion and decision making. The second is simply to provide an opportunity to make use of the task allocation charts which have to be produced anyway before job stream charts.

The cross referencing is an attempt to reduce the competition between the design teams and the user teams. It establishes a physical link between the two sets of documents, and to help both teams to be more informed about the workings of the other. The production of the cross references gives both the design and user teams the opportunity to discuss the overall design, and to understand each other's role in the DIADEM methodology. By appreciating the important role each group has in the design process, there is no need to compete with each other or to try and dominate the project.

The task catalog is a way to assist the construction of the charts by formulating users ideas about job design. The idea behind the development of a task catalog is similar to that of the function catalog in SSADM. By showing the design teams this catalog, they can see that how the user teams develop their documents, and that the techniques and procedures used to develop the charts are similar to the ones they use to produce data flow diagrams. They should then have the confidence to use and refer to the documents during discussions. Once they know more about each other's work, they can work together more easily and avoid confrontation because of lack of knowledge about each other's work.
A better designed training programme can help to co-ordinate the user teams members and the social consultants to work together, and to highlight to the team members the importance of the social and human aspects in systems design. If the users can be trained to realise the objective of JSC and to use it in a more effective way so as to only produce the relevant charts for system design purposes, then the over production of the charts will not occur.

There are reasons to suggest that there is a lot of redundancy between the job stream charts and data flow diagrams, which is especially noticeable during cross-referencing. Therefore, developing a single notation integrating both techniques together seems to be a sensible way forward. There is no reason why this cannot be done, and based on the illustrated examples in the earlier sections it is possible to achieve. However, producing such a single notation will undermine an important long term objective of the development of the technique of Job Stream Charting. Discussion with HUSAT personnel reveals that the ultimate aim is to have Job Stream Charting developed as a stand alone technique, capable of application in any methodology. Therefore, the Job Stream Charting technique should not be tied to a specific methodology. This means that one does not necessarily need to have an in depth knowledge of the chosen methodology in order to use Job Stream Charting. For example, if SSADM is to be used, then one does not need to know a great deal about Data Flow Diagramming in order to use Job Stream Charting. It is hoped that this will make it easier to have human factors technique built into any methodology that may not have adequate provisions for social considerations. Of course when Job Stream Charting is used in conjunction with another technique, then it is necessary to formulate and coordinate their integration.
Another reason for not creating a single notation to combine Job Stream Charting and Data Flow Diagramming is that both these techniques provide two particular views of the system. Data Flow Diagramming documents the different processes within the system, and the information flow between them. Job Stream Charting shows the task allocation and job specification of the system. These two views are necessary parts of the information to design and build the system. A single notation integrating the two techniques together will run the risk of compromising the objectives of both techniques, and therefore fail to provide the technical and the organizational data for the design process.

In the next Chapter, the issues of incorporating job design issues early in the systems design process will be discussed in a more general way, based on experiences of those involved the project using the DIADEM method, and the research undertaken in this project. The Chapter will conclude with a review of efforts made by HUSAT personnel to integrate human factors principles into structured design methodology.
5. Incorporating Human Factors Techniques into Methodologies

As we have seen (Checkland 1981 and Wood-Harper et al 1985) the social or human aspects of system design play an important part in producing a well designed and usable system. Suitable techniques should be provided to handle these social issues. However, when it comes to actually including these techniques into methodologies, problems start to appear. There are no clear guidelines on what these techniques are and how they fit in the design process; in what form the outputs from the techniques should be provided and how they can be used with outputs from other design processes; how the techniques can be used in harmony with those techniques that are technically oriented. The experiences of using job stream charting in DIADEM (Ip et al 1990) demonstrate the difficulties encountered in using such human factors techniques. In the previous Chapter, which dealt with the use of job stream charting, suggestions were made as to how to improve the use of the technique in future projects. This chapter aims to explore the possibilities of applying some of these suggestions at a more general level to the design process.

5.1 Management’s Recognition

The first point is the highlighting of the social and human issues in systems design as primary areas of concern, and thus should be treated with the same attention and importance as other issues, such as the technical design or the choice of hardware and software. Very often the design process is dominated by the technical aspects of the system (Ip et al 1990). The social and human issues are frequently treated as secondary or additional features and not as part of the fundamental requirements. This may be due to the
perceptions of management who ultimately make the decision to implement a proposed system in an organization. It seems that they are happier to accept advice from technical consultants on the more concrete issues such as the sort of system required for the business, which manufacturer's equipment should be used, the costs of installing and maintaining the proposed system. But often they do not seem to appreciate the advice given by human factors consultants. Consider for example the DIADEM project (Ip et al 1990). Here management would attend presentations organised by human science consultants to put across their points of view; they would also accept reports of analysis on subjects like the organizational impact of the proposed system, its possible effects on the staff and the general working practice of the business. Afterwards though, the management did not seem to know what to do with this information. They would tend to assume that the system would cope with the social and human requirements. Therefore, the whole concept of the social and human implications in a new system should be properly explained to the management, to ensure that it will continue to treat these issues as seriously as other areas of systems design.

5.2 Improve Output Quality

The second point concerns the outputs of the human factors techniques and the presentation of social and human issues. The problem is related to the use of words and terminology and the general style of presentation. This problem is similar to that often associated with technical specifications or analysis. The output from these activities is often expressed and illustrated in the "languages" familiar to those who produce them. Those who have to use them have difficulties in extracting the information they
require and hence fail to understand the significance and implication of the information. Therefore, efforts should be made to make the output of the techniques more understandable and usable to those who are expected to use them to assist decision making. Their content should be written in a clear, concise and simple manner with the minimum usage of specialist terminology. Diagrams should also be clear, with appropriate indices and explanatory notes where necessary. They should also be produced under control so that the required information is captured without over production of irrelevant material. Where possible, the authors should be available to give advice and clarification on any queries. In this way, any decisions by the management will be made with confidence based on the output of the analysis.

5.3 More Human Scientist Involvement

The third point concerns the usage of the human factors techniques, in particular, the amount of guidance and instructions provided to those who use these techniques. During the DIADEM project (Ip et al 1990), the human factors consultants adopted a "hands off" approach. They indicated what needed to be done concerning the social aspects of the design process, and then left the design teams to carry out the work. Often the teams did not know what to do at each stage, how to do it, and when they had actually completed it. Discussion with one of the consultants suggests that because the social and human issues and requirements of a new system are different for different organizations, it would be better to let the responsible design teams ascertain these requirements themselves. Then they can see to it that these requirements are incorporated into the final system. This is certainly a valid reason for the approach. However, without an appropriate understanding of what a method is for, why it
is likely to be useful, and how to apply it, people will not have the motivation to continue with it as soon as problems occur. Hence if the advocates of the method believe that a crisis of confidence will occur, then they must be involved in monitoring progress in order to provide guidance as required.

In the case of the DIADEM project and the use of the job stream charting technique, problems arose from the start. The design teams had a member of the user team attached to it who was supposed to oversee the development of the charts. What seems to have happened is that the design teams produced many charts by simply following the chart construction guidelines, without thinking what or why they were doing them. One of the aims of job stream charting is to represent the allocation of jobs and tasks in the current system. What seems to have happened is that charts were produced for every job in the organization, including those which were not really relevant to the system. Another aim of job stream charting is to represent different options in the job design and work organization in the proposed new system. What seems to have happened is that many more charts were produced showing different permutations and combinations of jobs and staff, apparently without much thought as to their suitability in the new system.

The suggestion here is that the human scientists ought to take a more active role in the design process. Rather than taking the remote approach, they should be involved more deeply in the appropriate areas of system design. The degree of involvement will be no more than that suggested in the participative approach discussed in an earlier Chapter. Although they should not be actually doing the design teams' work for them, they should make themselves more available so that they could answer questions from the team and give
suitable guidance to the teams when they are using the various techniques.

5.4 Summary

In this dissertation, the problems with system design was discussed, followed by the reasons for embarking on a proper system analysis and design process and the need for design methodologies. This was followed by the discussions on the different trends of methodology development: technical, social, combination (socio-technical) and others such as Soft Systems Methodology and prototyping. The discussion then moved on to the methods of comparing methodologies, followed by a report on such a comparative study. The dissertation then focussed on the need to incorporate human factors into methodologies and efforts to achieve this in DIADEM. In particular, the discussion concentrated on a technique called Job Stream Charting, its application and problems encountered. Improvements to the technique were suggested with illustration and a worked example.

There is no doubt that the social and human aspects of systems design are gaining more attention and prominence. And efforts by HUSAT to incorporate the associated techniques into the various methodologies had some success. In general, there are several areas of improvement that can be made. Firstly, the need for the management to realise the importance of the social implications of introducing a new system into the organization. Management should ensure that these areas are properly catered for from the beginning of the system design process. Secondly, the quality of output from these human factors techniques should be made more understandable and usable to management and other decision
making bodies. If the key points can be recognised and included in the basic system design, then as further components are built upon it the human factors issues will not be forgotten. Thirdly, the human scientists should be more directly involved in the design process, helping out with any problems experienced by the design teams, and to giving measured guidance on the usage of the techniques. They should be available to give constructive assistance. There are other suggestions for improving the practice of human factors techniques. They all work towards providing a better all round system development methodology which, it is hoped, will produce a more socially acceptable and usable computer system.
Appendix A : SSADM

SSADM stands for Structured Systems Analysis and Design Method. It was developed by Learmonth and Burchett Management Systems in conjunction with the UK government's Central Computer and Telecommunication Agency (CCTA). The methodology was accepted by the CCTA in January 1981. In January 1983 it became compulsory for government departments to use SSADM for all systems development work. The method has been frequently updated with version 3 released in July 1986 and version 4 released in September 1989.

SSADM is suitable for the development of medium to large systems. SSADM follows the "cook book" technique to help the teams to apply the methodology. Each step in the methodology is described in detail: the tasks involved, the techniques used, the required input and the expected output.

SSADM offers several design features which propose to help with the development work. SSADM is essentially data-driven. It uses techniques to capture the underlying and stable information handled within the system, how it flows into, out of, and around the system, and how the entities associated with the information change over time. SSADM actually requires cross-checking to be done at intervals to make sure that all areas of development work are done correctly and in a coherent manner. The logical and physical features of systems design are separated in SSADM. By doing this, the logical design can specify what an ideal information system would incorporate - e.g. a highly flexible, maintainable system. This will then be refined when the physical design is developed as a consequence of software availability, financial consideration, staff experience, and other physical constraints. Therefore SSADM
recommends that the transition from logical to physical design is done as late as possible.

SSADM does not provide any specific guidelines as to the composition of design and development groups who will be using the method for their work. It seems that within each design team, there has to be at least one expert SSADM user (possibly the systems analyst or an external consultant) and others with at least some knowledge of the method. The role of the management and staff of the organization throughout the development is not specified in SSADM and as such is a matter for design teams to decide.

How does SSADM work?

There are three phases in SSADM: feasibility study, system analysis, system design.

In phase one, the feasibility study phase, the business case and technical feasibility of the project are considered. Estimated costs and benefits, and relevant social considerations are taken into account to see whether the analysis phase should go ahead. This phase is not compulsory but is recommended for a more thorough and proper use of the method. There are two stages in this phase: problem definition and project identification. The two stages and their associated steps provide an "accurate" definition of the total problem to be addressed by the new system, and offer a set of options to tackle the problems defined in the previous stage, which are formalised into a feasibility report.

The problem definition stage seeks to provide an accurate definition of the total problem to be addressed by the new system. This stage has 6 steps:
Step 1 - initiate feasibility study: establish base constraints for study, create initial problem/requirement list, top level data flow diagram (DFD) and overview logical data structure (LDS), identify major areas for investigation, project planning, establish quality assurance (QA) review group.

**technique**: data flow diagramming, estimating, planning, LDS technique.

**input**: project terms of reference.

**output**: initial problem/requirement list, top level DFD, overview LDS, project plan.

Step 2 - create current system overview: investigate the current system by function area, construct second level DFDs, note down new problems and requirements, add function descriptions to DFDs, review second level DFDs with users, update top level DFD if necessary.

**technique**: data flow diagramming.

**input**: top level data flow diagramming, initial problem/requirement list.

**output**: current top and second levels DFDs, basic function descriptions, problems and requirements.

Step 3 - create overview data structure: refine and review LDS, create outline entity descriptions, note down new problems and requirements, review LDS with relevant users.

**technique**: LDS technique.

**input**: overview LDS, initial problem/requirement list.

**output**: updated LDS, outline entity description, problems and requirements.
Step 4 - develop logical system overview: using logicalisation techniques to current physical DFDs, note down new problems and requirements, amend LDS and entity descriptions, create datastore/entity cross-reference, revise elementary function descriptions, review logical DFDs and updated LDS with appropriate users.

Technique: DFD logicalisation, LDS technique.
Input: current system physical DFDs, LDS, basic function descriptions, entity descriptions.
Output: current system logical DFDs, updated LDS, problems and requirements, updated basic function descriptions, updated entity descriptions, datastore/entity cross-reference.

Step 5 - consolidate initial problem/requirement list: add problems and requirements from previous steps to initial list and resolve any duplication or combine appropriate points, then review list with user for accuracy and prioritises the list.

Technique: none.
Input: problem/requirement list, additional problem/requirements.
Output: consolidated and prioritised problem/requirement list.

Step 6 - review problem definition: distribute products to reviewers before holding a QA review, agreed on system boundary, incorporate any changes required by the review, review project plan, obtain formal approval for the next phase.

Technique: quality assurance.
Input: current system logical DFDs, LDS, problem/requirement list, function descriptions, entity descriptions, datastore/entity cross-
reference, project plan.

output : approved problem definition consisting of all products input, formal authorization document, amended project plan.

The project identification stage offers a series of options to tackle the problem defined in the previous stage, and they will be formalised into a feasibility report. There are four steps in this stage:

**Step 1** - identity outline project options: create base constraints for all options, outlines of a number of possible solutions, brief impact analysis for each option, review suggestion with user.

**Technique** : data flow diagramming.

**Input** : approved current system logical DFDs, LDS, problem/requirement list.

**Output** : two to four project skeletons consisting of top level DFD, outline hardware layout diagram, impact analysis.

**Step 2** - create outline project specifications: for each of the project skeleton from the previous step, create second level DFDs, LDS, functional descriptions, estimate sizes of entities, technical environment descriptions, high level project plan, develop impact analysis, cost/benefit analysis.

**Technique** : data flow diagramming, LDS technique.

**Input** : the project skeleton plans from the previous step.

**Output** : for each project, a specification containing top and second level DFDs, LDS with estimated entity sizes and volumes, functional descriptions, technical environment descriptions, high level
project plan, impact analysis, outline cost/benefit analysis.

Step 3  - evaluate project options: distribute copies of outline project option specifications to those responsible for evaluating the options, prepare formal presentation to compare each option, assist user in selecting one option, record the chosen option and the reason for selection.

technique: user option selection.
input: outline project specifications.
output: selected outline project specification and reasons for selection.

Step 4  - formalise feasibility study report: complete project specification, update outline cost/benefit analysis, carry out detailed estimates and costing for the analysis phase, create project plan for the analysis phase, the covering text for formal report, review report with responsible users, obtain formal authorization for continuation to the analysis phase.

technique: data flow diagramming, LDS technique.
input: all products from previous steps.
output: a formal feasibility study report covering:
   1) problem definition comprising of current system physical and logical DFDs, current system LDS, problem/requirement list;
   2) outline project option specifications;
   3) statement of reasons for selection of option;
   4) selected project specification comprising:
      - DFDs and function descriptions;
      - LDS and outline entity descriptions;
      - updated problem/requirement list;
Phase two is the systems analysis phase. It has three stages involving analyzing the current system, new user requirements and user problems with the current system. This phase becomes important if the feasibility study phase is not carried out beforehand. If no feasibility study is carried out first, then analysts and design teams will be encountering the total problem for the first time at the stage where they should be concentrating on detailed analysis work. They will not have the benefits of conclusions from the feasibility study, such as whether the project is actually viable, the definition of the problem to be addressed by the new system, and descriptions of the current system in the form of top level data flow diagrams. The overall project plan is another important document to have come out of the study. Without the study, all the groundwork for the project will have to be done at this phase, consuming more time and resource, and making this phase more significant.

Stage 1 is the analysis of system operations and current problems. This stage has five steps:

Step 1 - initial analysis: review contents of start up document, carry out initial overview analysis, agree initial scope with user/project management, identify users and areas for analysis, initiate problem/requirement list, create project plan for the analysis phase, get agreement for project plan, initiate project documentation and quality assurance mechanisms.

- technique: LDS techniques, data flow diagramming.
- input: analysis start up document.
- output: top level DFD, initial LDS, problem/requirement
list, project plan, interview plan, QA brief.

**Step 2**
- investigate current system: review and update top and second level DFDs, create three level DFD if needed, review or create elementary function descriptions, validate DFD processes against LDS, note down any problems.

**Technique:** data flow diagramming.

**Input:** top and second level DFDs, interview plan.

**Output:** current system physical top, second and third level DFDs, elementary function descriptions, problems/requirements.

**Step 3**
- investigate system data structure: redraw and extend LDS, create entity descriptions for each entity on the LDS or review entities if already existed, carry out a first pass relational data analysis, create a data dictionary and record entities and attributes, validate LDS against DFD processes, note down problems/requirements reviewed.

**Technique:** LDS techniques, relational data analysis.

**Input:** initial LDS, LDS and entity descriptions from feasibility study if produced.

**Output:** updated LDS, entity descriptions, problems/requirements.

**Step 4**
- develop problem/requirement list: review contents of initial problem/requirement list and expand each entity, insert any new problem/requirement.

**Technique:** none.

**Input:** initial problem/requirement list, noted new problem/requirements.

**Output:** problem/requirement list.
Step 5 - review investigate results: agree system boundaries with user, carry out QA review, resolve points raised and amending products as necessary, obtain formal authorization to continue stage 2.

**technique**: quality assurance.

**input**: QA brief, project plan, current system physical DFDs, elementary function descriptions, LDS, entity descriptions, problem/requirement list.

**output**: all input products, amended project plan, formal authorization to continue stage 2.

Stage 2 involves building on the user requirements expressed in terms of the problem/requirement list. Audit, security and control requirements are added in order to complete the required system specification. This stage has eight steps:

**Step 1** - define logical system: apply logicalisation techniques to DFDs, match LDS entities and DFD datastores and create datastore/entity cross-reference, amend function descriptions if necessary, note down problems, amend data dictionary if necessary.

**technique**: DFD logicalisation.

**input**: current system physical DFDs, elementary function descriptions, LDS, entity descriptions.

**output**: current system logical DFDs, datastore/entity cross-reference, function descriptions, problems.
**Step 2**  
- define audit security & control requirements:  
  identify requirements for fallback and recovery, access limitation, controls to be incorporated into the system, audit, criteria for handling exceptions, data exceptions.

  technique: none.
  input: current system logical DFDs, LDS.
  output: initial requirements for audit security and control.

**Step 3**  
- define and consolidate user requirements:  
  incorporate problems defined in this stage so far, audit security and control requirements into problem/requirement, review the list with relevant users, create constraints list if necessary.

  technique: none.
  input: problem/requirement list, problems, audit security and control requirements.
  output: expanded problem/requirement list, constraints list.

**Step 4**  
- identify and select from business system options:  
  if not created during feasibility, create up to six business systems options, discuss them with responsible users and narrow down to two or three, for each one create top level DFD, function descriptions, LDS, brief impact analysis and cost/benefit analysis, discuss further with users and assist them to select one option.

  technique: data flow diagramming, LDS technique, user option selection.
  input: problem/requirement list, current system logical DFDs, function descriptions, LDS.
  output: one selected business system option with
- further define chosen option: redefine top level DFD to second and third level DFDs where necessary, match required system logical DFDs with current system logical DFDs, modify lower level DFDs by incorporating solutions to problem/requirement list not already solved by the selected option and update problem/requirement list, create or modify function descriptions, create function catalogues which group functions into logical enquiry and logical update (on-line or batch) types, validate catalogues with LDS, update option documentation, datastore/entity cross-reference and data dictionary, create input/output specifications.

**Step 5**

- create required data structure: examine each entry on problem/requirement list and, if the solution is to be provided by the selected business system option, then any necessary changes are made to LDS and entity descriptions; complete entity descriptions and ensure volumes, key attributes and other known attributes are included; update datastore/entity cross-reference and data dictionary.
technique: LDS technique.

input: problem/requirement list, business system option.

output: required system logical LDS, expanded entity descriptions, updated problem/requirement list.

**Step 7** - investigate detailed system logic: create entity life history (ELH) matrix, event catalogue, create ELH for each entity, resolve inconsistencies revealed by ELH, update all products to reflect resolution of inconsistencies, create logical dialogue outline for each on-line event in the event catalogue, update problem/requirement list.

technique: ELH, dialogue design.

input: required system logical DFDs, function catalogues, problem/requirement list, input/output descriptions, datastore/entity cross-reference, elementary function descriptions, required system LDS, entity descriptions.

output: ELHs for each entity, logical dialogue outlines for each on-line event, updated required system logical DFDs, LDS, entity descriptions, problem/requirement list, function catalogues, input/output descriptions, datastore/entity cross-reference, ELH matrix, elementary function descriptions, event catalogue.

**Step 8** - review required system specification: review problem/requirement list, carry out QA review with all products, resolve points raised and amend documentation as necessary, update data dictionary, obtain formal authorization to continue stage 3.

technique: quality assurance.
input : all products form step 7.
output : agreed required system specification, formal authorization to continue.

In stage 3, users are invited, and assisted, later on in the last stage to choose a technical option for implementation of the required logical system, i.e. to select the required physical system. This stage has four steps :-

**Step 1**  - create technical options : create up to six high-level technical options, a list of constraints which the options must satisfy, reduce technical options to two or three based on technical feasibility, expand each into more detailed specification containing - a technical environment description, a function description, an impact analysis, and outline development plan, a cost/benefit analysis.

*technique* : first cut program and data design, user option selection.

*input* : all products from stage 2.

*output* : option specifications.

**Step 2**  - user selects from technical options : prepare presentation plan, prepare presentation(s), deliver presentation(s), provide follow up assistance and advice to users in option selection, note down decision made by users in selection.

*technique* : user option selection.

*input* : option specification, required system specification.

*output* : note of option decisions, option specification.
Step 3  - complete and review required system specification:
  update all products in required system specification to reflect chosen option, complete system specification of chosen option, review it with user, amend specification if points raised by users, create project plan for design phase, implement plan.

  technique: none.
  input: note of option decisions, option specification, required system specification.
  output: updated and completed required system specification, project plan for the design phase.

Step 4  - define performance objectives:
  specify data storage criteria, function timing criteria, information objectives, recovery criteria, any other performance criteria.

  technique: none.
  input: project plan for the design phase, required system specification, function catalogues, LDS, entity descriptions, problem/requirement list.
  output: performance objective specifications, all other products.

The final phase is the systems design phase. It involves designing the required system both logically and physically using the specifications provided from the analysis phase. There are three stages in this phase: data design, process design and physical design.

The data design stage involves design of the data structures for the required system culminating in the composite logical data design. This is often carried out in parallel with the process design. There are two steps in this stage:-
Step 1  - conduct relational data analysis: select the sources (normally those shown as input to this step) for relation data analysis, collect samples and specifications, create relational data analysis plan, carry out the analysis on each source, optimise the results, update data dictionary with the results.

    technique: relational data analysis.
    input: input/output descriptions, entity descriptions, current system file structures.
    output: optimised third normal form (TNF) relations, updated data dictionary.

Step 2  - create detailed logical data design: create data structure from TNF relations, merge structure with LDS to form composite logical data design (CLDD), complete entity descriptions to reflect CLDD structure, update data dictionary to reflect CLDD.

    technique: CLDD.
    input: TNF relations, LDS, entity descriptions, function catalogues.
    output: CLDD, completed entity descriptions.

The process design stage is carried out in conjunction with the data design stage. There are three steps in this stage:

Step 1  - define logical enquiry processing: create logical enquiry process outline for each function listed in the retrievals function catalogue, create output formats for each output identified on the enquiry process outlines, update data dictionary, update logical dialogue outlines with CLDD.
technique: process outlines, dialogue design.
input: retrievals function catalogue, CLDD, function descriptions for enquiry functions, input/output descriptions, dialogue outlines.
output: logical enquiry process outlines, output format, logical dialogue outlines and controls.

Step 2 - define logical update processing: update event catalogue with events identified in CLDD, create ELH on entities in CLDD but not in LDS, remove ELH for entities not appear in CLDD, review remaining ELHs and amend as necessary, amend other products of the required system specification due to updating of ELHs, update logical dialogue outlines to become consistent with CLDD, create logical dialogue outline for new events from updating ELHs, create logical update process outline for each event in event catalogue, update CLDD, entity descriptions, and data dictionary with the creation/updating of logical update process outline.

technique: ELH, process outlines, dialogue design.
input: update function catalogues, CLDD, event catalogue, ELHs, input/output descriptions, logical dialogue outlines.
output: logical update process outline, updated ELHs and entity descriptions, CLDD, DFDs, function descriptions and function catalogues, logical design outlines.

Step 3 - validate and review logical system design: plan and carry out a QA review of all relevant products, amend logical system design and data dictionary as result of the QA review obtain formal authorization to continue to stage 6.
In the physical design stage, the logical design from the previous two stages is used to build the required physical system. Once the detailed design work is accepted, the detail of the final system specification is created. There are nine steps in this stage:

**Step 1** - create first cut physical data design: obtain first cut design rules (from reference manual) for the appropriate database management system (DBMS), apply these rules to CLDD to give first cut physical data design, produce file content or page plans and update data dictionary to reflect first cut design.

- technique: first cut physical data design.
- input: CLDD, entity descriptions, data dictionary.
- output: first cut design data design, updated entity descriptions and data dictionary.

**Step 2** - create program specifications for major transactions: designate each process outline as batch or on-line, create batch and on-line program specifications.

- technique: first cut program specification.
- input: products from the previous step, from stages 4 and 5; required system logical DFDs, function catalogues, CLDD, logical update and enquiry process outlines, datastore/entity cross-reference, input/output formats and descriptions.
output : first cut program specifications consisting of physical program descriptions, menu and dialogue descriptions for on-line programs, run flows for batch programs, physical process descriptions, transaction file definitions.

Step 3 - create performance production and tune design: select programs or transactions which will dominate system performance, calculate the timings for each program or transaction, modify physical design products produced so far to meet performance objectives not so far cater for, get agreement from users to modify if performance objectives cannot be met, modify physical design products due to the revisions, ensure program timings can be met in the revised physical design, update data dictionary with the revisions.

technique : physical design control.

input : first cut physical data design, first cut program specifications, performance objective specifications.

output : modified physical data design, modified first cut program specifications, modified entity descriptions.

Step 4 - create file/database definitions: create detailed record specifications for the DBMS, create file specifications and/or database schema.

technique : none.

input : physical data design, entity descriptions.

output : file specifications and/or database schema record specifications.
**Step 5**

- Complete program specifications: produce system runcharts showing all programs and files in each processing cycle; create detailed specification of each program consisting of cover page, program runchart (batch) or schematic diagram of processing (on-line), narrative program description, physical process description for each process/transaction in the program, specification of all input/output formats, specification of files/records/data items accessed, timing forms for each transaction, list of common facilities/routines to be used; update data dictionary, define system building schedule, create a test plane for each program, create a project plan for the programming phase, carry out QA review of products of this step.

**Technique:** quality assurance.

**Input:** physical data design, program specification.

**Output:** system flowcharts, detailed programming specifications, updated data dictionary, system construction plan, program test plans, project plan for programming phase.

---

**Step 6**

- Create system test plan: create plan for link testing the integrated system, plan for testing the system to meet user requirement, plan for system acceptance testing, a project plan for the testing phase, carry out formal QA review of products of this step.

**Technique:** quality assurance.

**Input:** required system specifications, problem/requirement list, file/database design, all products from previous step.

**Output:** test plans as specified.
Step 7  - create operating instructions: create operating instructions for batch and on-line programs.
technique: none.
input: program specifications, physical data design.
output: operating instructions as specified.

Step 8  - create implementation plans: create data conversion plan including description of conversion method; outline of special conversion program required, specification of content and validity tests required to ensure clean data is taken, project plan for conversion tasks; create implementation plan, review plan with relevant users.
technique: none.
input: required system specification, program specifications, test plans.
output: implementation phase plans.

Step 9  - define manual procedures: refine required system DFDs, define final screen formats, describe in detail on-line dialogue including interactive conversations, enquiry/report messages, system messages, access authorization procedures; define in detail methods of input where applicable, e.g. input creation and data entry, error messages and reports, resubmission of rejected transactions; create back up and recovery facilities, create operating schedule, create end-user facility instructions, consolidate all products in a user manual.
technique: none.
input: operating instructions, required system specification, physical data design.
output: system user manual.
Appendix B: ETHICS in Detail

ETHICS stands for Effective Technical and Human Implementation of Computer-based Systems. This is a participative design approach which was developed by Professor Enid Mumford of Manchester Business School. Its principle objective is the successful integration of company objectives with the needs of employees and customers. It is based on the suggestion that good systems design is more than just good technical design, but also the organizational context in which the technology is placed. It provides a systematic methodology and set of analytical tools which analyses the efficiency and job satisfaction needs, set design objectives to improve these needs, and develop strategies to achieve these objectives.

Two groups of participants are set up who will have monthly meetings. The Steering Committee consists of senior management from the user, systems development, finance and all other major areas, including senior union officials where necessary. It sets guidelines for the Design Group. The Design Group consists of representatives of all those interested in the design area as well as the professional systems analysts, and will meet once a week or a fortnight. It will design the system: the choices of hardware and software, man-machine interactions, the overall reorganization of the design area and the allocation of responsibilities and tasks to groups and individuals. Its terms of reference and its level of participation must be agreed before the project begins. The roles of the systems analysts, the facilitator, and departmental managers must also be clearly defined. Both groups will then have to learn the ETHICS approach as well as other interested parties from the management to trade unions.
Apart from the principle objective, ETHICS has four other main objectives:–

1) To base system design on an accurate and careful diagnosis of business problems and human relations needs.
2) To give equal weight to these problems and needs.
3) To ensure that the design task encompasses good organizational design as well as good technical design.
4) To create systems which are effective, efficient, acceptable and stimulating.

The following is a summary of the fifteen-step breakdown of the ETHICS methodology.

Step 1 establishes why the existing system needs changing and what needs to be changed. This is discussed amongst the Design Group members. Minutes of discussion and proceedings are then circulated to the Steering Committee and all staff in the user area.

Step 2 identifies the system boundary where the Design Group's responsibilities begin and end. It has to design in detail the system within the boundary which it has identified and to ensure that the system links easily and effectively into functions which adjoin it and into the external environment. This is then discussed with the Steering Committee. System boundaries are likely to change from the start of the project, and any such changes should be clearly understood by both groups.

Step 3 constructs a detailed description of the present system. ETHICS requires two kinds of description. The horizontal input/output analysis describes the documents and information coming into a department, what is done to them, and what documents and information come out of the
department. The vertical analysis shows the different levels of work complexity and importance. The activities are analyzed at five levels: the operating activities which should have been described in the input/output stage; the problem prevention/solution describe activities at preventing and correcting work problems; The co-ordination activities describes those activities that have to be co-ordinated within the department and between departments; the development activities describes those activities, products and services that needs to be developed and improved; The control activities describes how the total department is controlled so that it works efficiently, meets its targets and achieves its objectives.

Step 4 sets out to define the key objectives which the design areas should achieve. Two questions are asked about every area within the design boundary:

1) Why do they exist and what is their primary role and purposes?
2) Given this role and purpose what should be their responsibilities and functions?

Afterwards a third question can be asked: How far do their present activities match what they should be doing?

Step 5 seeks the answer to the question: What are the key tasks which must be carried out if the key objectives are to be met? No details need to be included at this stage.

Step 6 identifies the key information needs associated with the key tasks. This can be based on the five-level model used in step 3.
Step 7 looks for variances which affect the efficiency needs. There are two types of variances. Key variances are potential problem areas which cannot not be eliminated but can be effectively controlled. They usually appear at the interface with other systems. Operating variances are weak links unintentionally built in by earlier systems design, which can be eliminated when the new system is installed. The gathering of this variance information provides the opportunity for members of the Design Group to discuss with the staff they represent, and thus get all the staff involved in the design of the system.

Step 8 tries to obtain job satisfaction needs, which is given equal importance with efficiency under ETHICS. It covers three board areas. Needs associated with personality, including knowledge needs and psychological needs. Needs associated with competence and efficiency in the work role, including efficiency factors and user needs. Needs associated with employee values such as ethical needs, treatment of employees by the management and between themselves. This information is obtained by the ETHICS questionnaire, which should be handled by the facilitator who is in a neutral position. After analysis, the results are discussed within the Design Group. The Steering Committee will also have a copy as well. The Design Group members then discuss the results with their constituents. Any suggestions are fed back to the Design Group which fills out an analysis of social needs form. This form helps the group to incorporate the good job aspects into the new system and to improve those that are not.

Step 9 identifies and analyses future changes that are likely to affect the system within the next five years. These changes include the available technology, legal
requirements, economic factors, employee and customer attitudes, company organization.

Step 10 is a key step in ETHICS. The efficiency, job satisfaction, future change problems and needs identified in the previous steps are listed out. These should be stated positively as objectives to be striven for through the way the new system is designed. Each member of the Design Group will rank the objectives in order of importance with regard to the constituents they represent. Others not represented in the group should be asked to take part, as well as external groups such as customers and suppliers if possible. The results are then examined closely. Any conflict is resolved, although sometimes it is left alone if a satisfactory resolution is not possible. Then a priority objective list is made and ranked. They should be checked with the key objectives and key tasks in steps 4 and 5, and must be agreed by all members of the Design Group. They are then discussed with the Steering Committee and other interested groups to make sure of their approval. This avoids rumour and misconceptions generating. These objectives are then translated into specific and measurable targets.

Step 11 concerns the organizational design of the new system. The Design Group is moved on from diagnostic activities to learning design skills, as well as the various technical options. The group will eventually come up with the organizational options (different ways of organizing the department to achieve the objectives). Information from steps 3, 4 and 5 are used and answers to the following questions should provide the basis for the organizational design of the system :-
1) Given the definition of key tasks in step 5 what are the operating activities that must be carried out?

2) What are the problem prevention/solution activities that will result from the key objectives and key tasks which are identified?

3) What co-ordination activities will be required within the design area and between it and other area?

4) What development activities will be necessary to keep the new system adaptable and in a constant state of improvement?

5) What control activities will be necessary? What kind of targets will need to be set and how can performance best to be monitored?

6) Given this complex mix of activities are any special skills required in all or some of the user area staff?

7) Are there any key roles or relationships to which particular attention must be paid in the design of the new system?

Each organizational option should specify the following:

1) A broad outline of the organization of the design area as a whole in terms of large work groups, or sections, and their responsibilities.

2) A more detailed description of the work of each large group or section in terms of sub-groups and their responsibilities.

3) A description of how sub-group tasks and responsibilities will be distributed amongst small teams and individuals.

It is advisable for the Design Group to develop between three and six organizational options. Each option is checked against the efficiency, job satisfaction and future change.
objectives listed in step 10. Any advantages and disadvantages are noted.

Step 12 is the development of the technical options, including hardware, software and man-machine interface. These options are developed in the same way as the organizational options, including the evaluation. Where possible, experimental examples should be set up to help with the evaluation. Then a short list of both sets of options are drawn up. The group will then try to find a suitable combination of organizational option and technical options. Any such combination has to be evaluated against the priority objectives and those of particular interest groups. The best fit will be selected for detailed design and implementation. This will then have to be confirmed with the Steering Committee and the Design Group constituents.

Step 13 starts the preparation for the detailed work design. This requires the use of flow charting, recording the flow of information and documents through the design area. Responsibilities must be allocated and tasks distributed to groups and individuals so as to provide an efficient work environment. Particular attention must be paid to the creation of effective relationships and procedures across the system boundary with adjoining and interacting departments.

Different tasks associated with the new system are divided up into a number unit operations. The mix of tasks with each unit operation is selected to meet job satisfaction needs of the different kinds of staff working there. This kind of group structure gives individuals the support of a small, helpful team of colleagues, and enables tasks within the unit operation to be allocated to meet personal preferences. It also provides excellent training for new
staff who can begin with the routine work and progress to the more complex. All members of a group can learn all tasks and eventually become knowledgeable and multi-skilled. However, this mix of tasks should also comply with the following good job design principles:

1) There are clear boundaries between the different unit operations so that each work group has a feeling of identity with its work.

2) The sets of tasks contained in a unit operation provide a good mix of simple, intermediate and complex activities.

3) The work group is able to solve a majority of its own problems without having to refer them to a supervisor or another group.

4) The work group is responsible for the internal organization and co-ordination of its activities (who does what).

5) The work group is responsible for developing improved methods for carrying out its tasks. It may also be given other development responsibilities.

6) The work group can set many of its own targets and monitor its own performance in achieving these.

7) The work group can easily identify the targets which it has to achieve.

There are other important points for the Design Group to note. Any new organizational structure should be kept as flexible as possible so that it can easily changed. There are many ways of combining tasks together. The notion of Job and of job ownership should not be encouraged. Staff should be expected to constantly review the set of tasks for which they are responsible and requested, or be requested, that this be altered to meet new needs and new machines. It must also ensure that a high quality physical environment is created for those who will be working with the system.
Step 14 involves the implementation of the new system which must be carefully planned and executed. A strategy for the implementation can be developed by answering the following questions:-

1) What kinds of problems are likely to be encountered on implementation and how can these be avoided?
2) During the implementation period what activities will have to be co-ordinated, both within the design area and between other areas?
3) What training is necessary and how and by whom will this be provided?
4) How much time is required for implementation and how can progress best be monitored?

In the last step of ETHICS, the new system is tested to see whether it meets all the objectives set at the design stage. Improved efficiency and job satisfaction will also be checked.
Appendix C: DIADEM in Detail

The following are the stages, segments and worksteps involved in DIADEM, the Departmental Integrated Application Development Methodology (DIADEM 1987). Worksteps with * indicate that they are carried out in parallel.

Stage One - Initiation

The project is mobilised and the groundwork laid for further development. There are five segments:-

Segment One - Project Initiation
Workstep 1 - Appoint Project Steering Committee
Workstep 2 - Prepare Initiation Stage Plans
Workstep 3 - Organize Project

Segment Two - Outline User Requirements
Workstep 1 - Survey Current System
Workstep 2 - Define Outline Data Requirements *
Workstep 3 - Define Outline Functional Requirements *
Workstep 4 - Identify User Acceptance Criteria *
Workstep 5 - Assess Organizational Impact

Segment Three - Plan, Cost and Report
Workstep 1 - Prepare Specification and Logical Design Stage Plan
Workstep 2 - Prepare Feasibility Study Report *
Workstep 3 - Confirm Business Case *

Segment Four - Model Physical System
Workstep 1 - Develop Model General Structure
Workstep 2 - Develop Workload Profile *
Workstep 3 - Perform Sensitivity Analysis *
Workstep 4 - Produce Procurement Requirements

Segment Five - Management Appraisal
Workstep 1 - Conduct End Stage Review

Segements Three and Four are carried out in parallel. The major outputs from this stage are Procurement Requirements and Feasibility Study Report.

Stage Two - Specification and Logical Design

An agreed User Specification and a complete Logical Design of the new system is produced.

Segment One - Organization Implementation
Workstep 1 - Organize Project

Segment Two - Detailed User Requirements
Workstep 1 - Review Current System Survey
Workstep 2 - Define Detailed Data Requirements
Workstep 3 - Define Detailed Functional Requirements
Workstep 4 - Define Detailed Inputs and Outputs
Workstep 5 - Define Detailed Process Outlines
Workstep 6 - Define Operational Acceptance Criteria
Workstep 7 - Produce Job Design and Work Organization

Segment Three - User Review
Workstep 1 - Produce User Specification
Workstep 2 - Perform User Review

Segment Four - Installation Requirements
Workstep 1 - Define Installation Strategy
Workstep 2 - Define Data Conversion Strategy
Workstep 3 - Assess Impact on the Organization
Segment Five - First Procurement Requirements Confirmation
   Workstep 1 - Confirm Procurement Requirements

Segment Six - Education and Training Requirements
   Workstep 1 - Define Education and Training Strategy

Segment Seven - Logical Design Schedule Confirmation
   Workstep 1 - Confirm Logical Design Plans

Segment Eight - Logical Design Completion
   Workstep 1 - Refine Logical Processes *
   Workstep 2 - Refine Logical Data Structure *
   Workstep 3 - Confirm Logical Interfaces

Segment Nine - Conversion Logical Design
   Workstep 1 - Define Logical Conversion Processes

Segment Ten - Plan, Cost Report
   Workstep 1 - Prepare Physical Design Stage Plans
   Workstep 2 - Prepare Full Study Report *
   Workstep 3 - Confirm Business Case *

Segment Eleven - Second Procurement Requirements Confirmation
   Workstep 1 - Confirm Procurement Requirements

Segment Twelve - Management Appraisal
   Workstep 1 - Conduct End Stage Review

The following segments are carried out in parallel: Three, Four, Five and Six; Eight and Nine; Ten and Eleven. The major outputs from this stage are User Specification, Operational Requirements and Full Study Report.
Stage Three - Physical Design

A physical design for the new system and the conversion system is produced and documented.

Segment One - Organization Implementation
   Workstep 1 - Organize Project

Segment Two - Technical design
   Workstep 1 - Confirm System Architecture
   Workstep 2 - Design First Cut Database *
   Workstep 3 - Perform Physical Design Control *
   Workstep 4 - Develop Technical Specification *
   Workstep 5 - Confirm Physical Interfaces *
   Workstep 6 - Produce Technical Specification Report *

Segment Three - Conversion Design
   Workstep 1 - Develop Conversion Specification

Segment Four - Third Procurement Requirements Confirmation
   Workstep 1 - Confirm Procurement Requirements

Segment Five - Build Approach
   Workstep 1 - Define System Build

Segment Six - Test Strategy
   Workstep 1 - Define test Strategy

Segment Seven - Plan, Cost and Report
   Workstep 1 - Define Education and Training Strategy

Segment Eight - Management Appraisal
   Workstep 1 - Conduct End Stage Review
The following Segments are carried out in parallel: Two, Three and Four; Six and Seven. The major outputs from this stage are Technical Specification Report and Memorandum of Agreement.

**Stage Four – Development**

The tested system is produced for implementation.

**Segment One – Organization Implementation**

Workstep 1 - Organize Project

**Segment Two – Detailed Design**

Workstep 1 - Complete technical Design *
Workstep 2 - Establish Development Environment *
Workstep 3 - Design Programs *
Workstep 4 - Prepare Outside Test Plan and Data *

**Segment Three – Conversion Preparation**

Workstep 1 - Conversion Detailed Design
Workstep 2 - Conversion User Manuals Development *
Workstep 3 - Conversion Program Development *
Workstep 4 - Conversion Operations Guides Development *
Workstep 5 - Conversion Training Preparation *
Workstep 6 - Conversion System Testing *

**Segment Four – Program Development**

Workstep 1 - Develop Program Code
Workstep 2 - Prepare Inside Test Plan and Data
Workstep 3 - Perform Unit Test

**Segment Five – Development Installation**

Workstep 1 - Install Facilities for Development
Segment Six - User Manuals Development
   Workstep 1 - Produce User Procedures
   Workstep 2 - Produce User Documentation

Segment Seven - Acceptance Test Preparation
   Workstep 1 - Confirm User Acceptance Criteria
   Workstep 2 - Prepare Acceptance Test Data

Segment Eight - Operations Guides Development
   Workstep 1 - Establish Operations Environment
   Workstep 2 - Develop Technical Instructions *
   Workstep 3 - Develop Operations Procedures *

Segment Nine - System Test
   Workstep 1 - Create Application and Installation Test Plans *
   Workstep 2 - Prepare String Tests Plans and Test Data *
   Workstep 3 - Prepare Application and Installation Test Data *
   Workstep 4 - Perform String Tests *
   Workstep 5 - Perform Application Test *
   Workstep 6 - Perform Integration Test *

Segment Ten - Plan, Cost Report
   Workstep 1 - Prepare Installation Stage Plans
   Workstep 2 - Prepare Release Package *
   Workstep 3 - Develop Installation Test Plan *
   Workstep 4 - Develop Site Schedule *
   Workstep 5 - Confirm Business Case

Segment Eleven - Training Preparation
   Workstep 1 - Develop Training Programme
   Workstep 2 - Develop Training Material
Segment Twelve - Acceptance Testing
Workstep 1 - Run Acceptance Test

Segment Thirteen - Management Appraisal
Workstep 1 - Conduct End Stage Review

Segments Two to Twelve are carried out in parallel. The major outputs from this stage are: Program Specifications, User Manuals, Operations Guides, Training Programme, Acceptance Test Report.

Stage Five - Installation

The new system is implemented in the user environment. The application and conversion systems are installed and personnel are trained to use them. A Technical Review Report is produced after implementation.

Segment One - Organization Implementation
Workstep 1 - Organize Project

Segment Two - Implement Data Capture and Maintenance Release
Workstep 1 - Data Capture and Maintenance Training *
Workstep 2 - Install Facilities for Data Capture and Maintenance *
Workstep 3 - Install Data Capture and Maintenance Release *
Workstep 4 - Create and Maintain Data Capture Files

Segment Three - Implement release
Workstep 1 - Train Personnel *
Workstep 2 - Install Facilities for Release *
Workstep 3 - Install Release *
Workstep 4 - Convert Files *
Workstep 5 - Conduct Technical Review

Segment Four - Plan, Cost and Report
  Workstep 1 - Prepare Operation Stage Plans
  Workstep 2 - Confirm Business Case

Segment Five - Management Appraisal
  Workstep 1 - Conduct End Stage Review

Segments Two, Three and Four are carried out in parallel. The major output from this stage is Technical Review Report.

Stage Six - Operation

The development project is concluded and the responsibility for future operation is determined and assigned. The performance of the new system is monitored and changes necessary to tune the system or to correct off-specification items are tested and implemented.

Segment One - Organization Implementation
  Workstep 1 - Organize Project

Segment Two - System Assessment
  Workstep 1 - Tune System *
  Workstep 2 - Correct Off-Specification Items *
  Workstep 3 - Perform Post Implementation Review

Segment Five - Management Appraisal
  Workstep 1 - Conduct End Stage Review

The major output form this stage is the Post-Implementation Review Report.

174
Bibliography


Health and Safety Executive (1986) "Working with VDUs."


The National Electronics Council (1983) "Human Factors and Information Technology."


"PC User" fortnightly magazine, England.


Trade Union Congress (1985) "Guidelines on VDUs."


