A Decision Making Information System for Labour Resource Allocations: Integration of Logic and Time Based Large Scale Project Computer Simulations

Christopher Gerald John Stevens

Thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

Department of Systems Science
City University
London
United Kingdom

August 1991

In memory of my late Aunt, Aileen

Brevis esse laboro, Obscurus fio
[I struggle to be brief, and become obscure]
Horace
(65–8 BC)

Concepts which have proved useful for ordering things easily assume so great an authority over us, that we forget their terrestrial origin and accept them as unalterable facts. They then become labelled as 'conceptual necessities,' '*a priori* solutions,' etc. The road of scientific progress is frequently blocked for long periods by such errors. It is therefore not just an idle game to exercise our ability to analyze familiar concepts, and to demonstrate the conditions on which this justification of their usefulness depends.

Albert Einstein
(1879–1955)

Contents

List of Figures

List of Tables

Acknowledgements

> It is not so much our friends' help that helps us as the confident knowledge that they will help us.
>
> *Epicurus*, 3rd Century, BC

I would like to thank the following, who, over the last three and half years, have helped me with this research.

My research supervisor Professor Robert Flood of the Department of Management Systems and Sciences at the University of Hull, for identifying the seed and value of my ideas; through the many highs and lows encountered during the weary travels of postgraduate research it is necessary to have a supportive supervisor. Professor Ewart Carson my co-supervisor at the Department of Systems Science, for his support and ability to smooth troubled waters.

My great friends: Salim Filali for the tremendous time and help during the last two years and later stages of my work; Philip Doyle and George Lenoudias, for being constant encouragers, providing a sounding board for my ideas over the years and being steadying influences when the wheel kept falling off. Doctors Christopher Pope for his words of wisdom and Abdul Roudsari.

My gratitude to Steve Peterson of High Performance Systems, (producers of *Stella* for the Apple Macintosh), Lyme, New Hampshire, United States of America, for providing information on System Dynamics/computer simulation in project management. Gail Bowen of Micro Planning International, Westbury–on–Trym, Bristol for the loan of *X–PERT* and David Bowen of Claris Corporation, Stockley Park, Middlesex for the loan of *MacProject II*, both project management programs for the Apple Macintosh. These programs were used to validate information produced by the PERT within *PROMISS*.

The many other friends and people, too numerous to mention, who have given me encouragement and support.

The Science and Engineering Research Council for awarding me with a Postgraduate Research Studentship (Grant number 87314307).

Nomenclature and Glossary

α	Non–negative parameter.
Activity	An operation or process consuming resources and time.
Activity–on–arrow network	A network where activities are symbolised by arrows.
Activity–on–node network	A network where activities are symbolised by nodes.
Activity chain	A number of activities forming a sequential process.
AI	Artificial Intelligence.
A_t	PERT Actual or Activity Time.
Arrow	A connecting line showing a process direction between two nodes.
Aspect ratio	When referring to a network. The ratio, with respect to the number of activities along the critical path against the number in a parallel path any any one point (in time).
β	Non–negative parameter.
BIOS	Basic Input/Output System.
cfd	Cumulative Frequency Distribution.
CP	Critical Path.
CPA	Critical Path Analysis.
CPM	Critical Path Method.
CPU	Central Processing Unit.
Critical path	A path of activities with zero float. The total time of these activities determines the shortest time a project can be completed.
D	Maximum difference between the two cumulative distributions in the Kolmogorov–Smirnov test.
DBMS	Data–Base Management System.
Dependency	An arrow in an activity–on–node network that represents a sequence and arrow interrelationships of project activities.
$\partial f/\partial x$	Partially Differentiating f with respect to x.
DR.DOS	Digital Research Disc Operating System (a substitute DOS for use instead of MS/PC.DOS).
DSS	Decision Support System.
DT	Delta Time (change in time i.e., Δ).
DTO	Dockside Test Organisation.
DTP	Desk Top Publishing.
Dummy	An activity that neither consumes resources or time. It is used solely to provide a logical link between nodes.
Duration	Estimated or actual time required to complete an activity.
DYNAMO	DYNAmic MOdelling.
DYSMAP	DYnamic System Modelling and Analysis Package.
DYSMOD	DYnamic System Model Optimiser and Developer.
Event	A defined state during the project completion when all preceding activities are complete, but before the start of any succeeding activities.
ES	Expert System.
4GL	Fourth Generation Language.
f	Function.
F	Frequency, i.e., number of cases.

$F_0(X)$	Under H_0, the proportion of cases in the population whose scores are equal to or less than X. This is a statistic in the Kolmogorov–Smirnov test.
FF	PERT Free Float. Time by which an activity may be delayed or extended without affecting the commencement of any succeeding activity(s).
Float	Activity time plus any additional duration (which can also be negative) before the commencement of any succeeding activity(s).
GIGO	Garbage In, Garbage Out.
GUI	Graphics–orientated User Interface.
H_0	Null Hypothesis.
H_1	Alternative Hypothesis, the operational statement of the research hypothesis.
HAS	Human Activity System.
HSM	Hard Systems Methodology.
i	PERT Activity tail event.
IF	PERT Independent Float. Time an activity can be delayed or extended without preceding or succeeding activities being delayed.
IS	Information System.
ISS	Information Support System.
IT	Information Technology.
.J	DYNAMO notation for past time interval.
j	PERT Activity head event.
.K	DYNAMO notation for present time interval.
k	In the Kolmogorov–Smirnov test, the number of observations which are equal to or less than X.
Key–date	A date within a project that has to be achieved without restraining preceding activities.
K–S test	Kolmogorov–Smirnov test.
λ	Project Completion Time.
.L	DYNAMO notation for future time interval.
LM	Line Manager or Management.
Loop	An error in a network which results in a preceding activity imposing a logical restraint on an earlier activity.
LR	Labour Resource.
LRA	Labour Resource Allocation.
LRV	Law of Requisite Variety.
LSP	Large Scale Project.
μ_t	Mean time.
MIS	Management Information System.
MS.DOS	Microsoft Disc Operating System (used in non–IBM Personal Computers).
MSM	Mathematical Simulation Model.
MSoI	Meta–System of Interest.
$\wedge$	Most likely.
Multi–project scheduling	Resource allocation techniques to schedule more than one project by: a) considering one project at a time and scheduling all activities within the constraints of available resources; or b) considering all projects together and scheduling activities in priority order within the constraints of available resources.

Negative total float	Time by which the duration of an activity (or path) has to be reduced in order to permit a restraining key–date to be achieved.
Network	A diagrammatic representation of activities and/or events showing their interrelationships and dependencies.
Network complexity	Ratio of activities against the number of nodes.
NCP	Non–Critical Path.
Node	A point in a network at which arrows start and/or finish.
Non–splittable activity	An activity that once started must be completed without removal of resources.
OCP	Optimised Critical Path
OK	Optimal Knowledge.
OOPs	Object Oriented Programming.
OR	Operations Research.
Parallel scheduling	Activities are ranked in a priority order using a constant rule, grouped into start times. Allocation of LRs are made in priority order at each time period. If an activity cannot be started it is moved to the top of the succeeding time period group.
Path	A network arrow or arrows showing a continuous sequence of events.
PC	Personal Computer.
PCD	Project Completion Date.
PC.DOS	Personal Computer Disc Operating System (used in IBM Personal Computers).
pdf	Probability Density Function.
PERT	Program Evaluation and Review Technique.
Precedence network	Similar to an activity–on–node network where arrows show precedence relationships between activities.
Precedence relationship	An activity acts as a restriction because one activity must precede another activity, either in part or in total.
PROMISS	PROject Managers Information Support System.
PTG	Project tension graph.
PM	Project Manager or Management.
RA	Resource Analysis.
RAM	Random Access Memory.
R&D	Research and Development.
Resource	A definable variable required to complete an activity. Classified as either: a) non–storable: A resource which if not used in the period of availability does not accumulate for use in preceding periods; b) storable: A resource that is continuously available until exhausted.
Resource aggregation	Total resource requirements for each time period.
Resource allocation	Technique for scheduling activities and their resource requirements so that predetermined availability constraints are not exceeded (this also includes time).
Resource factor	Ratio of the sum of the project resources required against the maximum required at any one time given a predefined network condition.

Resource limited schedule	Activities are scheduled so that predetermined resources are never exceeded and that project duration time is minimised.
RW	Real World.
RWS	Real World Situation or System.
ROM	Read Only Memory.
Serial scheduling	Activities are prioritised and known LR requirements allocated against each one. Each activity is ranked in a serial format, taking into account the priority in which an activity has to be completed and LRs available. For example, if five labour units are required for any one activity, then this will not be started until sufficient LRs are released from a completed activity. Early start times are moved to the right in the network until sufficient LRs are available.
SD	System Dynamics.
SDR	Search Decision Rule.
SDM	System Dynamics Methodology.
Slack	Calculated time span within which an event must be undertaken.
$S_N(X)$	In the Kolmogorov–Smirnov test, the observed cumulative step function of a random sample of *N* observations.
SoI	System of Interest.
STELLA	Structural Thinking, Experiential Learning Laboratory with Animation.
STM	Square Ternary Matrix.
τ	Time constant.
TA	Time Analysis.
TF	PERT Total Float. Time an activity can be delayed or extended without affecting the total project duration.
Time limited scheduling	Activities are scheduled so that key–dates and project duration is not exceeded.
2DM	Two Dimensional Matrix.
t_m	PERT Most likely time.
t_o	PERT Optimistic time.
t_p	PERT Pessimistic time.
σ^2	Variance.
WIMP	Windows, Icons, Mouse, Pulldowns.
WSoI	Wider System of Interest
X	In statistics, this means any observed score.
$X > Y$	X is greater than Y.
$X \geq Y$	X is equal to or greater than Y.
$X < Y$	X is less than Y.
$X \leq Y$	X is equal to or less than Y.
$X \in Y$	X is an element of Y.

Abstract

This thesis is about: A Decision Making Information System For Labour Resource Allocations: Integration of logic and time based large scale project computer simulations. The aim is to develop a rigourous means of enhancing computerised project management information systems, for use with labour resource allocations. This is achieved by integrating a network schedule with a mathematical simulation model to produce PROMISS (PROject Managers Information Support System).

The broad structure has three main focus areas: a) background to the topic; b) current position and direction for development; and c) implementation and effect of analysis of the integrated model.

As PROMISS is to be used in a large scale project (LSP) environment this thesis starts with a discourse on how such operations are managed. A central part of planning and controlling these projects is use of Program Evaluation and Review Technique (PERT) network schedules. A chapter is devoted to the theoretical and mathematical foundations of this technique and a PERT program developed in the following chapter.

This research thesis advances the use of System Dynamics, computer simulation and decision support systems for use in LSP environments. Making decisions relating to labour resource allocations directly influences success or failure of a project, but current information systems do not facilitate such decisions To help project managers, a mathematical simulation model (as a subjective reflection of a situation) is introduced to anticipate what *may* happen in the future.

Once various scenarios have been simulated a new logic plan is produced for those carrying out the tasks. Integration of information between a PERT and simulation model provides a new logic plan of actual activity start and stop times and their duration. PROMISS, in summary, starts with an original PERT then information it provides is simulated to take into account environmental influences, not originally taken into account when the project was planned.

This thesis provides a discourse about a number of issues related to the development of PROMISS and how it enhances decision making and managing large scale projects. PROMISS enables Project managers to make quicker and better informed decisions.

Preface

Inspiration—the Start

Genius, it is said, is the ability to grasp the obvious.
Anonymous

The manager wants information not facts, and facts only become information when something is changed. The manager is the instrument of change....
Stafford Beer

Prior to 1984 when joining the Department of System Science at City University as an undergraduate, I was an engineering manager involved in the refit of Hunter/Killer nuclear powered Fleet submarines for the Royal Navy. Refit of such boats is both diverse and complex, involving almost every facet of engineering from repairing simple switches to refuelling a nuclear reactor. During this time I realised that management and control of these projects was extremely complex. Management control and planning (as in all similar projects) is almost totally reliant on computer software based on an idea conceived in the late 1950s. During the last thirty years use of such software has hardly changed, though hardware and software developments used in other managerial situations have advanced beyond recognition.

Today, almost all large scale projects (LSP) concerning civil and industrial engineering use similar planning methods, within a project management environments. Reliance on such computerised planning systems have increased availability of *data, information* without a corresponding increase in the amount of useful information available.

This research thesis is an accumulation of theoretical and practical ideas developed from my eighteen years of industrial experience and six years at university. My industrial experience lead to a comprehensive knowledge and understanding of how a PERT network schedule worked and was utilised in the control of complex industrial projects. Whilst at work a problem existed in using this management information system for use in labour resource allocations (LRA), although I was unaware then of alternative methods that could have been used. In retrospect, it was found that such a system also restricted the project manager from considering his charge in a holistic manner.

During these studies I began to believe that there may be a more effective and efficient means of using computers, to help project managers complement currently used information systems. The basis of such a system was utilised in two dissertations (Stevens, 1986; 1987). As a result of this work Robert Flood wrote a paper with me (Flood and Stevens, 1987) outlining the concept of feedforward control in the management of LSPs.

This work and paper laid the foundation and motivation for a research programme to help project managers with decision making and so anticipate results of their actions, *before* allocation of labour resources. My experience showed that project managers currently only use feedback control using information derived from data that was, at best, several days old. An additional problem with such complex computer programs is their ability to provide a situation of "data *and* information overload." A constant problem for managers in a control situation, is to disseminate data and information before it can be used in the decision making process. This is invariably undertaken when there is an intense amount of pressure, including severe time constraints. Making a decision in an environment that may influence the final project completion date could, I believe, be made easier.

A project manager's information system needs to provide, from available data, guidance for two different groups of people. First, a logical guide for those carrying out the work and secondly, those planning and directing it. I knew from experience of working in LSPs, that only the first aspect was properly addressed.

A shortfall in providing suitable decision making information is covered up by the production of a vast array of report generation methods, currently provided by complex software programs and powerful computers. Such reports are produced from a few basic pieces of data, based on the same logic guide drawn out during the planning period, prior to commencement of the project.

I concluded that various aspects of computer modelling used in other environments had progressed over the last thirty years, leaving project management with only its original, but still much needed, logic guide of how to carry out the work. Today, the use of project management environments/ organisations is being increasingly adopted to manage projects of all sizes—this is likely to continue for the foreseeable future, with a greater reliance on computers to help in the control process.

My research objective, therefore, is to:

> Analyse the use of computer models (logic and time based) and improve the currently used information systems (PERT) in the management and control of labour resources in LSPs. Enhancement will be by making use of an integrated mathematical simulation model based on System Dynamics.

During the first twelve months of my research a review (Stevens and Flood, 1988) was made of past and current work in the field of project management information systems. The following two and half years were used to develop a prototype information support system—the result is PROMISS (PROject Managers Information Support System).

Some confirmation of the value of my work occurred in January 1990. An American industrialist at the *Society for Computer Simulation* Winter Multi-conference in San Diego, California asked me: "Why has this never been done before—it seems such a simple and clever idea to resolve an age old problem of conflicting requirements, found with currently used project management software?" I have continually asked myself this question over the past three and a half years. My research thesis provides an answer.

Chapter 1–*Large Scale Project Planning*, looks at how LSPs are planned, controlled and current information systems used. Chapter 2–*Program Evaluation and Review Technique—PERT*, details how this logic based model works and forms a foundation for PROMISS discussed in Chapter 3. Chapter 4–*Computer Modelling*, considers important aspects of time based mathematical computer simulation modelling. Chapter 5—*System Dynamics*, takes this specific Hard Systems Methodology and explains some of the theoretical foundations, practical advantages and limitations.

Chapter 6–*Modelling a Project*, details how various points discussed in the previous two chapters are implemented in context of modelling a project. This chapter is probably the most subjective aspect involved in this research, that of modelling a project. Consideration is given to the concept as a whole, that of developing a prototype alternative information support system.

Chapter 7—*Discussion—Feedback and Feedforward Control* considers these two forms of control and how they affect project environments. The discussion continues by looking at computer based information systems and PROMISS—the advantages and limitations. Chapter 8—*Integration—PROMISS* discusses how the two computer models are integrated. Chapter 9—*Conclusion* summarises research findings and provides future direction. As

PROMISS is a prototype system it is necessary that this last chapter discusses various limitations (in its developed form) but in this it provides a foundation for continued development.

Christopher G. J. Stevens
City University
London
United Kingdom

August 1991

Chapter 1

Large Scale Project Planning

> The moral is simple: one cannot specify what information is required for decision making until an explanatory model of the decision process and the system involved has been constructed and tested. Information systems are subsystems of control systems. They cannot be designed adequately without taking control into account.
>
> Russell Ackoff
> *Management Misinformation Systems*, 1967

1.1 INTRODUCTION

This research is directed towards the development of a large scale project manager's computerised information system for use in the allocation of labour resources (LR). Before such a system can be conceptualised, a brief discourse is required to understand and clarify how most large scale projects (LSP) are controlled. Organisational problems inherent in a two–dimensional matrix (2DM) management control structure (or any form of multi–dimensional structure) is not the subject here, but it does have an influence on the information system (IS) required for effective management in such an environment.

This chapter considers what a LSP is, how it is managed and what the IS requirements are:

> It is important to understand how the elements of an organization function as a system because, as in any other system, the organization operates through the medium of information. It is also necessary to understand that because the functions of management are served by an information system, these functions should be considered in the design process. (Murdick *et al.*, 1984: 368)

1.2 LARGE SCALE PROJECTS (LSP)

Previous experience of the author has been in managerial positions within a matrix project organisation, involving both project and line functions of refitting nuclear powered Hunter/Killer submarines and surface warships. This type of work constitutes a LSP, the planning and control being typical of projects such as the building of power stations or oil rigs. The work is a totally integrated exercise of a complex (many interrelated or interdependent activities) nature and in the case of submarines is carried out within a confined and hazardous environment, therefore necessitating good planning and controlling. In such projects, activities can number in excess of 6,000, taking 45,000 man-weeks with multiple Critical Paths (CP) over a two year period to complete. Orchestration of such work has to be well planned by the Project Manager or management (PM) involved. Many organisations have introduced matrix management designs to manage such projects.

> The current generation of management has developed two new forms as a response to high technology. The first is the free-form conglomerate; the other is the matrix organization, ... The matrix organization grows out of the organizational choice between project and functional forms, although it is not limited to those bases of the authority structure. (Galbraith, 1971: 29)

In parallel with the development of more complex LSPs, together with new management control structures, has been an increased use of computerised planning systems, (Moder *et al.*, 1983).

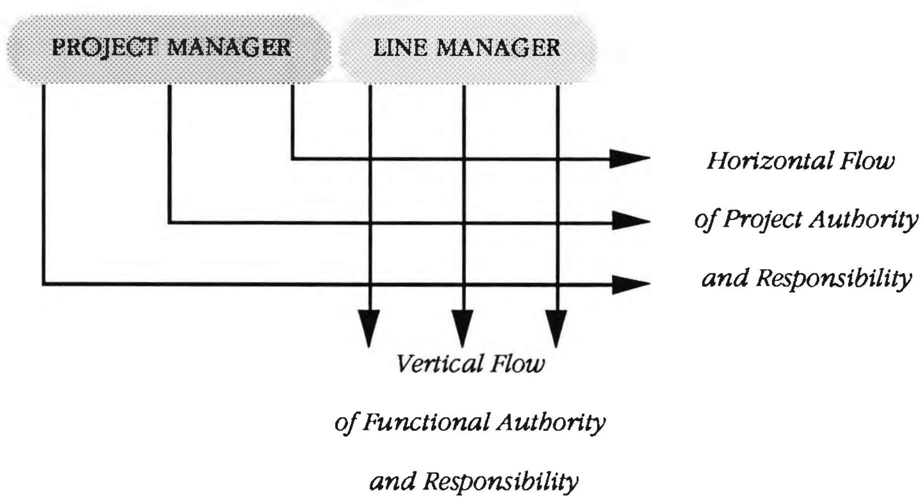
1.3 The Two Dimensional Matrix (2DM) Management Structure

Conventional management structures generally form a pyramid shaped organisation divided into functions, each with their own hierarchy of managers. Integration between common managerial levels within each function is limited to informal links, with formal communication channels at higher levels only. Simon (1957) considered organisations as being divided into two modes of specialisation: one vertical and the other horizontal. The former divides labour according to performance tasks. Such division is a power structure, based not on the task, but on the power to make decisions. Labour at the bottom of the organisation has little, if any, decision making authority (see Stevens and Wharton, 1991 for differences between authority and accountability). Those at the top do not generally have any direct involvement in the performance tasks, that is, they are directors of subordinates only. A 2DM organisation attempts to reduce the height of the power structure and mesh it with a labour structure undertaking the task. Division of labour and unity of control has been the subject of many theories and works since Adam Smith's manufacturing of pins, in his book *The Wealth of Nations*, (1776). It is not considered necessary to review these theories here.

In an attempt to reduce the level at which functions are integrated, a matrix form of management structure is formed for each project or product. This enables (in theory) decision making to be made at management levels closer to the practical aspect of completing a project, (Kast and Rosenzweig, 1985; Moder *et al.*, 1983; Parkin, 1980), '...to reduce the distance between the sources of information and the points of decision.... The purpose was to make as many decisions as possible at low levels with the people most knowledgeable.' (Galbraith, 1971: 32)

A 2DM also reduces the levels of communication links:

> The project team illustrates the principle of the matrix structure, which is characterized by formalized lines of lateral communication superimposed upon the separate vertical hierarchies of departments. Matrix structures in this respect formalize the informal lateral communication that would normally exist between departments and upon which many organizations rely heavily to keep themselves running smoothly. (Child, 1984: 98)

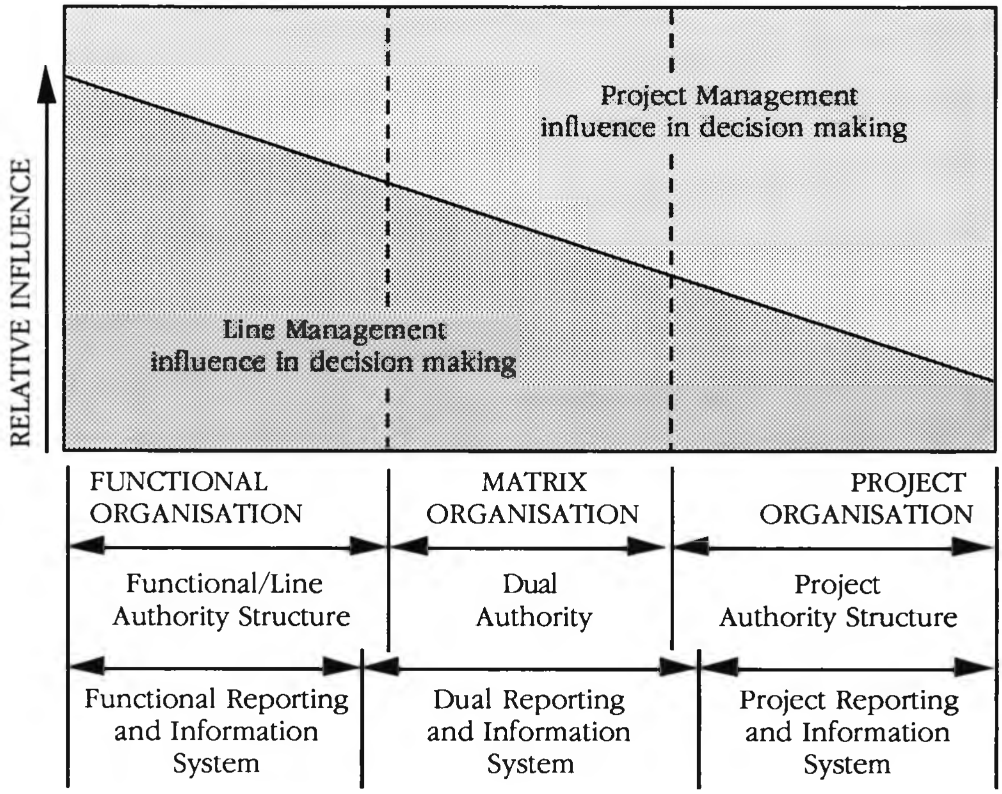


Typical two dimensional management matrix for a LSP
Figure 1.1

A matrix structure made of two lines of LSP management forming a 2DM organisation, see Figure 1.1. The two dimensions represent PM and Line Manager or Management (LM). This 2DM structure has been the most widely used means of controlling projects of any size. Most literature written on projects has been in the area of Research and Development (R&D), leading to product production (inception through to implementation), and in the construction industry. The former is where the PM and a small staff are assigned to the development and/or production of a single product, or as an example in the latter case, the construction of a civil engineering project. R&D projects will probably utilise staff and directly assigned LRs from functional (Line) departments within the parent organisation. A construction project is more likely to utilise new Line staff, under the control of the parent company's PM and key LM/supervisors.

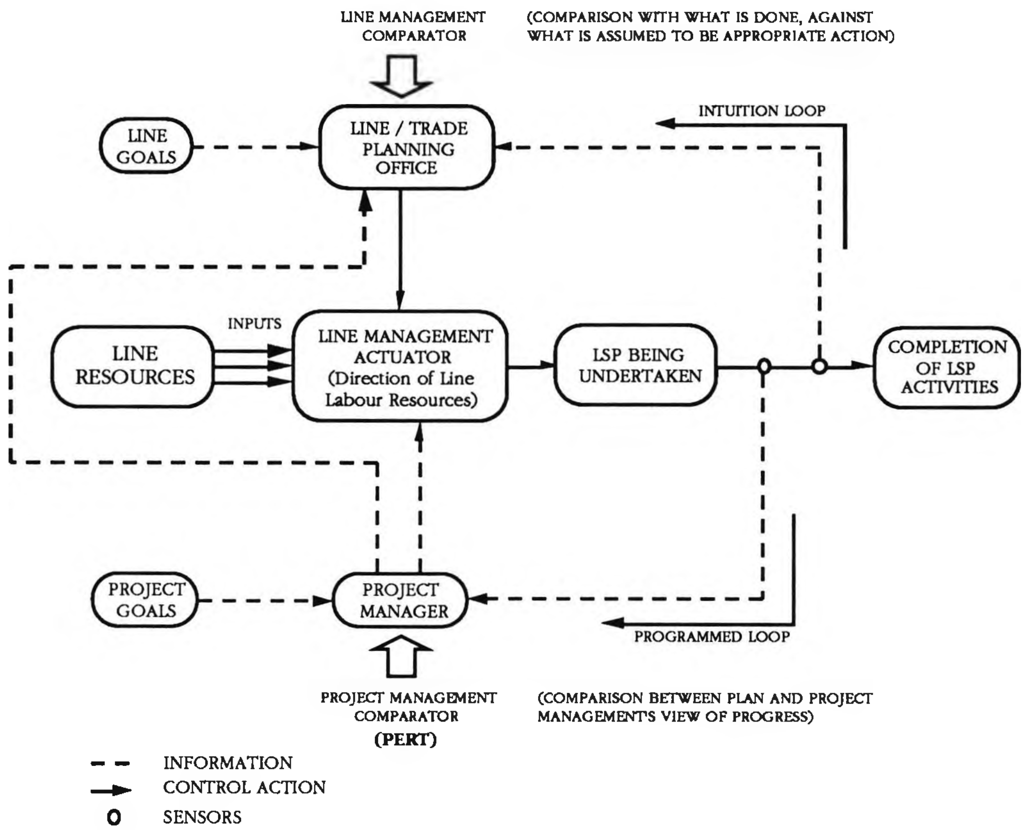
There is some doubt over the effectiveness of such divisions of labour, as Davis and Lawrence (1977: 7–8) said after an extensive review of such organisational structures: 'If you do not really need it, leave it alone. There are easier ways to manage organisations.'

Authority, reporting and information system requirements for functional, matrix and project organisations are shown in Figure 1.2, an adaptation of Galbraith (1971: 37).



Authority, reporting and information system
requirements for functional, matrix and project organisations
Figure 1.2

A simplified control system is shown in Figure 1.3. This identifies two control loops: one used by LMs based on "intuition" and the other "programmed" and operated by PMs. The first loop arises since LMs are in close contact with the work being undertaken on a day–to–day basis. They are aware of the PERT logic (see Chapter 2), but 'it is/can be' circumvented as work progresses, this is especially true when labour resource allocations (LRA) change and other work gains a higher priority. This loop involves comparison of what is *done*



Double loop control structure of a LSP
Figure 1.3

against what is *assumed* to be appropriate action. The second loop involves overseeing *what is actually carried out* and comparing this against *what is planned.* Recommendations and planning/schedule advice is then given to LMs to redirect/concentrate on critical tasks within the LSP.

PMs, in a number of cases, find themselves in a position of *data* overload, rather than one of insufficient *information.* In these situations there is a shortage of optimal knowledge (OK) for a decision to be made with a degree of certainty that it is achievable (Stewart, 1971). Problems exist in trying to disseminate the required information (OK), or even being able to ask the right questions (Stevens and Flood, 1988). Hammond (1971) identified the difficulty of people interpreting too much information. It was suggested that computer facilities with information feedback presented in a graphical format with interactive control should be provided. Balke *et al.*, (1973) and Hammond and Brehmer (1973) considered easier comprehension of information by people, if the relationships between a control variable and a possible causal variable are shown by visual display together with the provision of a description. As project organisations make use of computerised planning systems, more time and money appears to be spent collecting data and not enough on collating it.

To make the 2DM function as conceived, there needs to be a total commitment to effective coordination, (Galbraith, 1971: 29):

$$\text{Coordination} = f(\text{authority} * \text{information})$$

Authority or power (be it normative, coercive or utilitarian, i.e., positional) is embedded in the position a manager occupies within a 2DM structure. Information required is not so tangible; *what*, *how much* and *when* should it be available? Allison, (1971: 120) considered this within a national organisation, which may hold true for any size organisation:

> Information does not pass from the tentacle to the top of the organisation instantaneously. Facts can be "in the system" without being available to the head of the organisation.... Information must be winnowed at every step up the organizational hierarchy, since the number of minutes in each day limits the number of bits of information each individual can absorb. It is impossible for men at the top to examine every report.... But those who decide which information their bosses shall see rarely see their bosses' problem.

There is a need to filter out *data* not required for any one decision being made; but what *information* should remain and what should be excluded? Russell Ackoff (1967: B–148) states:

> ...the two most important functions of an information system become filtration (or evaluation) and condensation. The literature on MIS's [management information systems] seldom refers to these functions let alone considers how to carry them out.

Over twenty years later managers are still perplexed by this problem. Dissemination of information can be efficiently carried out by a computer, provided it is programmed to produce what is required at any one time—this is a human, rather than a software problem. The more information available, the more is likely to be asked for, hence a position of "overload" is quickly reached which also causes other problems:

> While we tend to think of boredom as arising from a deficit of stimuli (information underload), it also (and, in fact, more commonly) arises from excessive stimulation (information overload). Information, like energy, tends to degrade into entropy—into noise, redundancy, and banality—as the fast horse of information outstrips the slow horse of meaning. (Klapp, 1986)

In many managerial situations there probably is no right answer and no analysis method can ensure that good decisions will be made. An IS needs a method for checking presumptions which would otherwise be accepted, normally without question, so leading to the likelihood that decisions will be soundly based and carry with them credibility. To partially answer the question of what information is required, planning and decision making is classified into two broad terms, Satisficing and Optimising:

SATISFICING—Planner sets goals and objectives that are not too demanding i.e., they are achievable: '...human beings generally satisfice—look for adequate rather than optimal solutions to their problems.' (Simon, 1983: 22), i.e., one sets about to find a needle in the haystack, rather than finding the sharpest;

OPTIMISING—Planner tries to ensure that performance improves, by making maximum use of resources. (Operations Research, (OR) fits this criteria);

Efficient use of these types of planning are dependent upon the classification of behaviour patterns, either Deterministic or Stochastic, see Section 4.6.3.

The industry standard planning system for most projects of any size is a Program Evaluation and Review Technique (PERT) network schedule. Such a system, being logic based, assists LMs with their work schedule but is restrictive for PMs who are required to make LRAs (see Section 1.6).

1.4 CURRENT PRACTICE

PERT (with the Critical Path Method) network schedule (Gabriel, 1986) or such derivatives as Monte Carlo simulation techniques used for a LSP IS (Burt and Garman, 1971; Van Slyke,1963; Richman and Coleman, 1981) are generally computerised planning systems (Higgins, 1985). Considering the former here, this sequencing method is based on logic; the work is carried out in a logical sequence,. For example in the construction of a house, the foundations are laid before the walls are constructed. This logic is in a series format, but other work can be undertaken in parallel (in this example the making of window frames) to the main task(s) on the Critical Path(s)—work that is continuous for the entire length of the project.

PERT is described in greater detail in Chapter 2, but the strengths and weaknesses of an example system used in a LSP are given below, (Flood and Stevens, 1987: 210) derived from experience of the author:

STRENGTHS

1) It is an ideal logic guide for work to be undertaken;
2) work has to be planned;
3) critical paths of work are identified, provided the time estimates are accurate.

WEAKNESSES

1) It is not time based;
2) networks are complex and difficult to understand, consequently reduction into sub–nets is necessary;
3) The PM is unable to make feedforward labour resource allocation decisions;
4) excessive computer time is required for rescheduling (up to fifteen hours with a batch file system) which has to be undertaken overnight between 1700 and 0800 Hrs. with an *Internet 80S* PERT network schedule, running on a PRIME 750 mini–computer;
5) rescheduling time removes the possibility of using PERT as a decision making tool.

As this type of system determines how a LSP is organised, the PERT database requires to be updated and maintained throughout a project's life. Moder *et al.*, (1983), consider four reasons for this:

- Recording actual work completed (in part or completed activities);
- the network logic for future and uncompleted work may need to be rescheduled;
- rescheduling of remaining float; earliest and latest activity start dates;
- identifying, if any, new critical paths.

Item 1 is required for any information system that will be utilised for future decision making. If known information is not represented in the network then suspect answers will result from any subsequent computation, (i.e., Garbage In, Garbage Out—GIGO).

Items 2 and 3 are unlikely to be undertaken as frequently as ideally desirable (Weaknesses, Item 4 above). Rescheduling of such a LSP PERT is not readily undertaken at each reporting period: 1) because of the cost and length of computer processing time (although reduced by making use of desk top micro–computers, rather than mini or mainframes), but more importantly; 2) the time to plot new network drawings (even with the latest drum plotters) showing the new logic work sequence is prohibitive (a point made by Bittie (1962) as a limiting factor in the use of PERT). As a result of this, new CPs are not identified. Progress information is however monitored (against the last rescheduled network) in the form of project tension graphs (PTG), showing consumption of remaining float (see Section 2.5.2 for a full explanation of various floats).

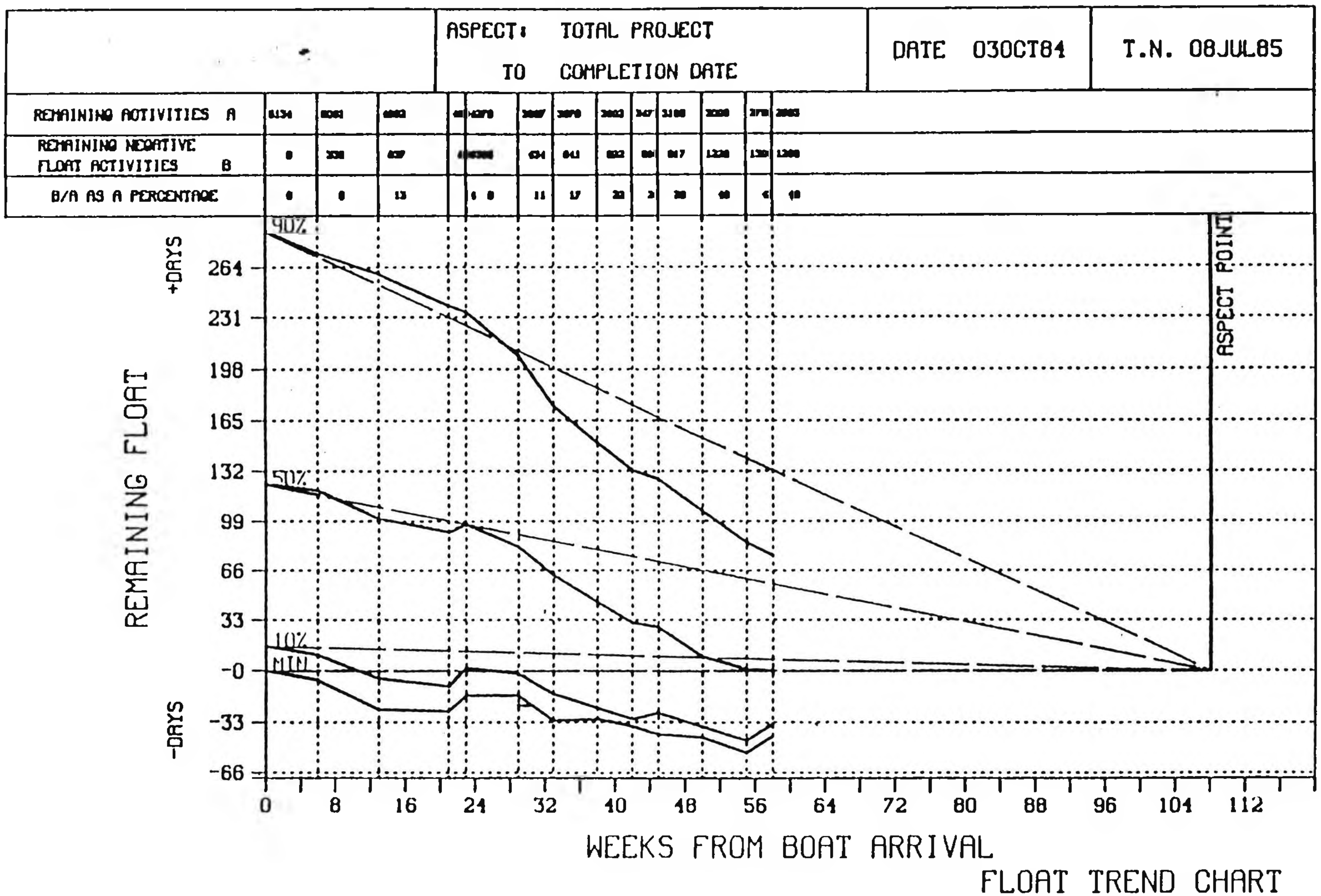


Figure 1.4
LSP tension graph

1.5 PROJECT TENSION GRAPHS

Remaining activity float values are put into descending order by the computer and the 90%, 50%, 10% and minimum 0% (being the CP) values located. Monitoring of how float is being consumed at each updating period, i.e., an increase in "tension," is determined by absorption of remaining float greater than planned. Figure 1.4 shows typical LSP PTGs. Broken lines meeting at the project completion date (PCD) show *planned* float consumption during the Project's life.

Dark lines represent *actual* float consumption, in this example showing a reduction of float values faster than planned. Persuasive power of PMs in bringing the LM back to the agreed plan is limited unless the float becomes negative (super critical), for example the 0% line in Figure 1.4 shows the original CP has fallen behind schedule. Some PERT paths, originally with float, i.e., non–critical, can become critical later when float consumption has been too high early in the project's life.

Another LSP health indicator is that of monitoring planned against actual activity, start and finish times. These are produced on a four line graph, see Figure 1.5 below.

1.6 MONITORING AND CONTROLLING

A key objective of the PM is to plan and then monitor progress of the project—as he and his staff have *total* responsibility (Lock, 1988). During completion of a LSP, as previously mentioned, PMs are constantly trying to anticipate potential problems and giving advice and information to LMs concerning "What" has to be carried out. PMs are primarily concerned with "How" the work is to be undertaken i.e., sequence of activities. Many decisions are made by managers who, when under pressure, normally respond by focusing on the present (Carlson, 1951; Mintzberg, 1973; Sayles, 1964), but many decisions are made by oversight and flight (Cohen *et al*., 1972).

A project manager's IS should consider differing requirements made upon it by users, whilst acknowledging that the organisation is unlikely to change its basic management structure (which may then need to change the IS requirements). Figure 1.6 shows an LSP hierarchy of plans originating from an original contract into operational plans to be carried out by the LM.

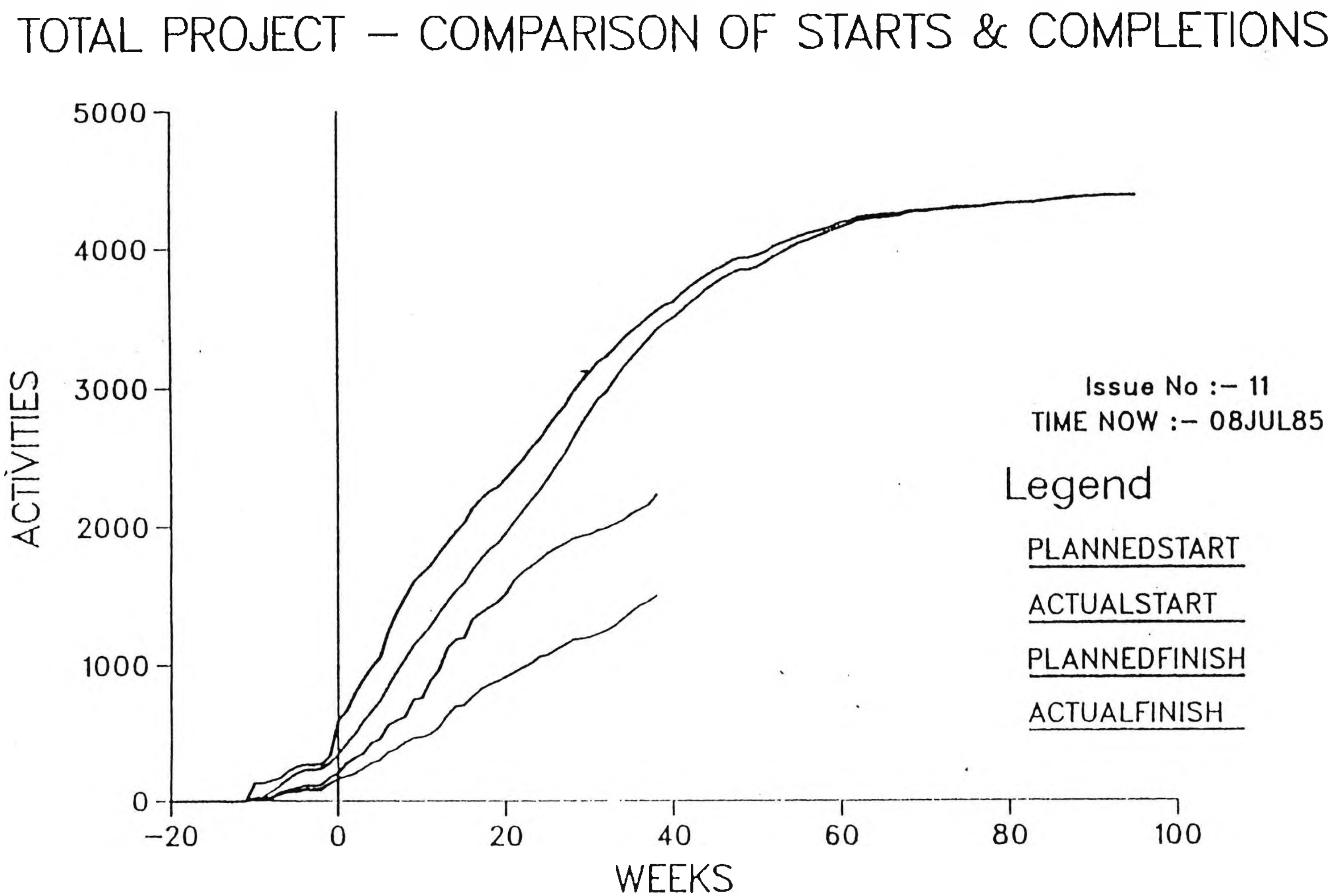
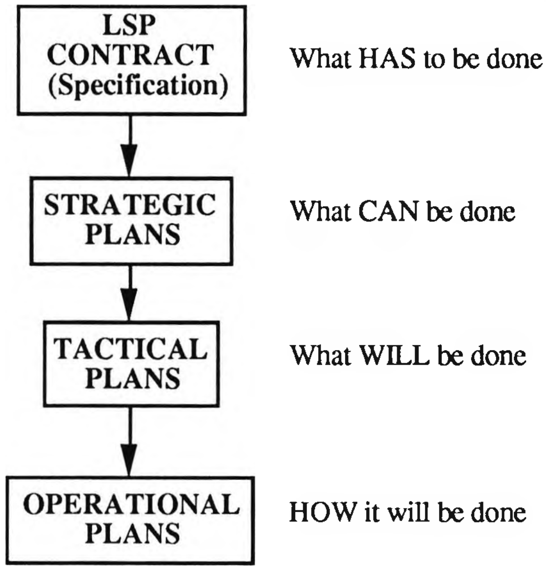
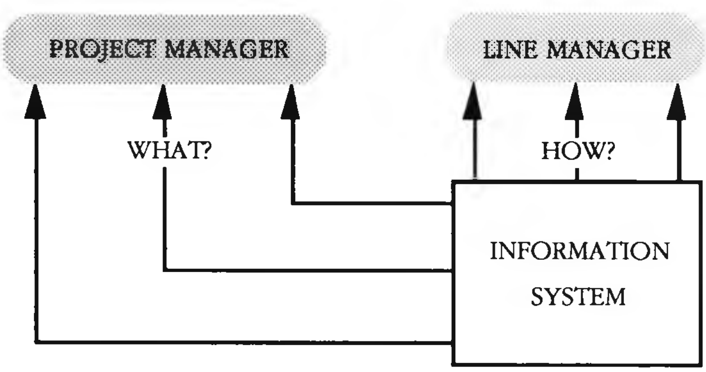


Figure 1.5
LSP activity progress graph



LSP hierarchy of plans
Figure 1.6

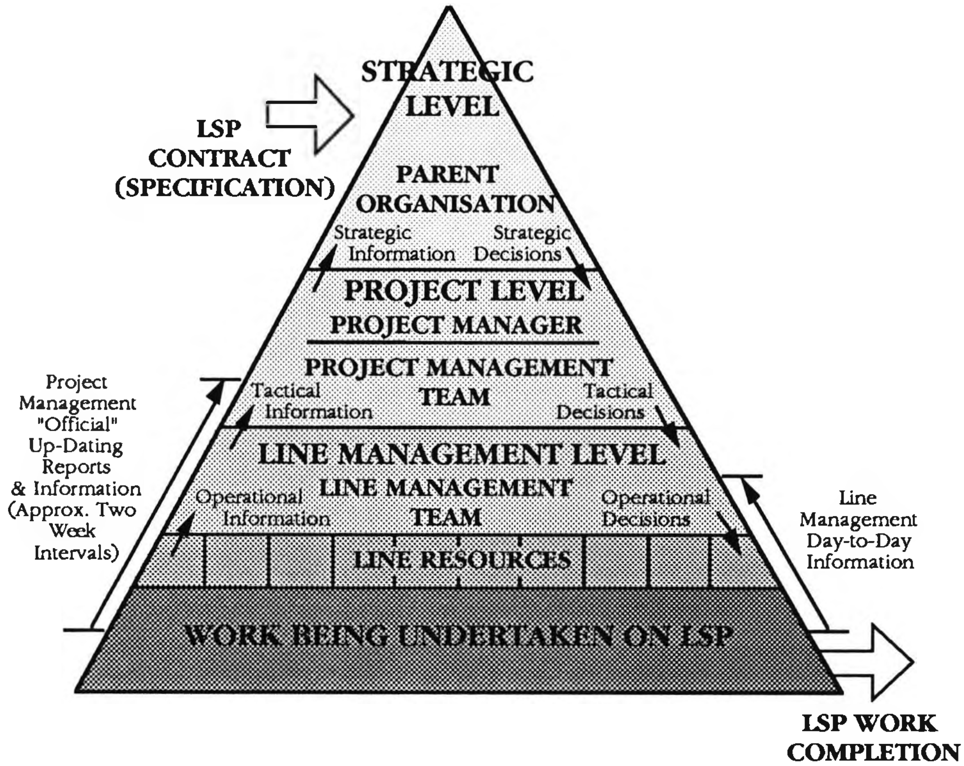
PMs are primarily concerned with *what* has to be carried out, with minimum cost and time to meet the customer's requirements. LMs, managers of labour resources undertaking the work, require a plan of *how* it is to be undertaken and in what priority order. Information requirements for each matrix arm are illustrated in Figure 1.7.



Information system requirement of a LSP
Figure 1.7

Figure 1.8 shows how information flows upwards as a response to decisions flowing downwards within a LSP organisation hierarchy.

As the PERT system is widely used in industry, it is generally understood by those who use it, without a requirement for a high level of technical knowledge, (unlike a simulation model for example) and is considered a good guide to those undertaking the work, namely LMs. Its use as a decision making system is however limited, due to the weaknesses given earlier. This is



Information/Decision flows within a LSP's management hierarchy
Figure 1.8

especially true when LRs are scarce (Berg, 1974; Blair, 1963; Muller–Merack, 1967). Many practitioners view current computerised practices poorly in terms of meeting the requirements of those operating within a LSP environment. Fox *et al.*, (1984) and Johanson (1986) suggested that computers should provide information (in the required format) for *all* those who need to use it.

A simulation model, however, is an easier and more effective means of analysing decision making scenarios and can be used to consider a LSP from a holistic, rather than reductionist, aspect (Battersby, 1970; Thimm, 1974). Although there is an increasing need for these computerised mathematical models, their current use is limited by the use of expert staff able to understand program algorithms and coding (Knuth, 1973).

1.7 Conclusion

This chapter briefly described current and most prevalent means of organising the management structure for a LSP, together with the use of information systems. Use of 2DM management structures may not be an ideal means of control, and is unlikely to change (in its basic formation) in the foreseeable future. Advancement of computer systems, both in hardware and software is considerable, but utilisation of this increased power and ability to manipulate data into information has, for those involved in LSPs, brought about a state of data/information overload.

PERT, in general terms, meets most information requirements for line management undertaking the work, but is far from ideal for project managers trying to determine how best to utilise labour resources. Integrating the two requirements to produce a single decision support system (as an information support system) offers a better means of providing necessary and required information to efficiently complete a LSP.

Chapter 2

Program Evaluation and Review Technique (PERT)

2.1 WHAT IS PERT?

Program Evaluation and Review Technique (PERT) was originally conceived in early 1958, by the U.S. Navy's Special Projects Department, to enable the Polaris Submarine Launched Ballistic Missile Project to be more easily controlled (Malcolm *et al.*, 1959, apparently the seminal paper; Freeman, 1960). This coincided with development of the Critical Path Method (CPM) produced by the DuPont and Remington Rand Univac Companies (Bildson and Gillespie, 1962; Kelley, 1961; Levy *et al.*, 1963). In parallel to these developments in the United States of America, the Central Electricity Generating Board of the United Kingdom conceived similar ideas when the expansion plan for building new power stations started. These planning methods were originally used as a scheduling system for the control of R&D teams and contractors involved in new projects.

Use of PERT network scheduling is ubiquitous within project orientated organisations (Lockyer, 1984; Murdick *et al.*, 1984) and as complexity increased, its use has become a necessity to plan and help with controlling LSPs. It is considered an industry standard planning system for projects of any size. Within four years of the Malcolm *et al.* paper there were seven hundred and two cited works related to this planning system (Dooley, 1964). When undertaking LSPs, the logic guide of PERT enables the Line function of a 2DM to carry out the work in an orderly and planned sequence. Logic can be predetermined by technical staff for implementation by others, who may be unfamiliar with planned or future progress of work.

This type of control system takes the form of identifying various activities involved in a project, estimating time needed to complete each one and then putting them in a logical order; one activity being preceded by another and succeeded by others, in an order normally required to complete the project.

PERT was developed for planning/scheduling when there are uncertainties (stochastic) in activity completion times i.e., determination of meeting the project schedule (Boulanger, 1961). The CPM system was originally used in situations where activity times were known, being founded upon a parametric linear program with the:

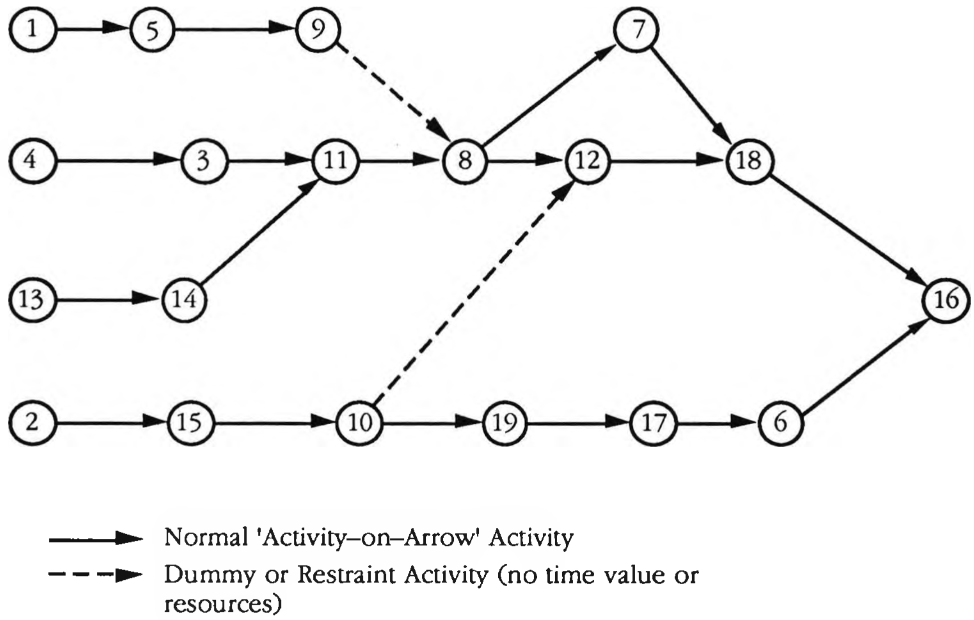
> ...objective of computing the utility of a project as a function of its duration. For each feasible project duration, a feasible project schedule is obtained that has maximum utility among all feasible schedules of the same project duration.' (Kelley, 1961: 297)

A logic based model such as PERT can also be considered as one of parts, not a whole with emergent properties unlike a mathematical simulation model.

2.2 COMPONENTS OF A PERT NETWORK SCHEDULE

A PERT network is a planning system where a project work schedule is broken down into activities (BS 4335, 1987; BS 6046, 1984). Simple networks consist of activities representing complete "parcels" of work or sub–assemblies. Larger and more complex networks represent more detailed work involved in completion of the project. Start and finish times of each activity are identified by an event node given a unique identification number (see Sections 2.4 and 2.5 for an explanation of this 'activity–on–arrow' method). Once all activities have been identified they are uniquely represented by a number determined by start and finish event nodes. Activities are then arranged so as to follow a logical sequence. This can be either in a parallel or serial format, in most cases network schedules are a combination of both, see Figure 2.1.

Considering the earlier example of the house construction in Section 1.4, the foundations are laid before the walls are built and the roof is put on. To save the overall construction time the window and door frames are made whilst the rest of the house is being constructed. Where two (or more) activities need to be undertaken in parallel for the completion of another future activity, an activity with zero time (and consumes no resources) called a dummy or restraint is introduced. This enables a computer to distinguish between the activities being worked on in parallel to each other. These are non–time valued activities connecting one logic path with another, i.e., an activity on one path cannot be started until all the preceding activities are completed from another path/direction, for example activity 1218 in Figure 2.1. Completion of sub–assemblies during the building of a large machine would be such an



Simple 'activity–on–arrow' network diagram
Figure 2.1

example. In the example of a submarine refit, there are over 6,000 activities, further complicated by nearly 3,000 dummies.

2.3 Critical Paths and Critical Path Method

This method makes assumptions that resources can be added and subtracted and then making an analysis with greater or less project (or activity) time and cost—this is known as a 'time–cost trade–off optimisation algorithm.' Differences between this and PERT systems (Phillips, 1964) have largely disappeared (most good PERT software packages are able to undertake CPM type of analysis) and distinctions are no longer made.

The longest path of activities that cannot be undertaken in any manner other than in succession (in series) to each other is known as the Critical Path (CP)—those logical sequences that are continuous throughout the project. This is the earliest time that a project can ever be completed and PERT determines this by use of Critical Path Analysis (CPA). In a LSP it is not unusual to find multiple CPs, in nuclear submarine refits these can number as many as six, compounding problems the PM has to deal with when LRs are limited (Clark, 1961b).

2.4 PERT NETWORKING FORMATS

Project planning using the PERT system utilises three alternative types of networking formats: 1) activity–on–arrow/link, or predecessor–successor event code (known as the $i \rightarrow j$ system where connections between events represent estimated or actual time); 2) activity–on–node; and 3) precedence diagramming. The first is the oldest and currently the most common format found in industry. The second is formulated by the same means, but a network is drawn with activity identification on nodes rather than connecting links, utilising the similar CPM format. This has found recent favour as it is easier to draw and follow.

Precedence method is more complicated to compute and is a development of the 'activity–on–node' system. This method splits up a network into greater detail to determine more exact relationships and logic of how activities precede others. This permits more activity overlapping and time lags involved in the work to be built in (Wiest, 1981). Increased sophistication of this format:

> ...also introduces some complexities of its own in the form of connecting arrows with several different definitions, and project time calculations that are not quite as straightforward as in arrow or node networks. (Moder *et al.*, 1983: 42)

Despite these limitations, this system is gaining favour within industry because of the ability to produce more accurate networks (in many cases one with a shorter project duration). Greater numbers of activity details required (and hence calculations) lead to an increased demand for computer processing power. Recent improvements in mini–computers have balanced out this disadvantage, for example the use of T and B's *Trackstar* system running on DEC VAX and the latest generation of Prime computers (this is in contrast to the rival mini–computer based system *Artemis* (almost considered an industry standard for projects of any size) which has been simplified to run on Personal Computers, see Ratti, 1986).

Formulation of PERT using the widely used notion of 'activity–on–arrow' format is discussed in the next section. From this theoretical foundation the logic based model within PROMISS is developed in the following chapter.

2.5 DETERMINATION OF 'ACTIVITY–ON–ARROW' ACTIVITY TIMES

2.5.1 INTRODUCTION

PERT has limitations in that activity times used in LSPs are generally stochastic in nature. Using a Beta distribution to calculate the mean, variance and activity independence is beset with many assumptions, thus, an *accurate* network schedule is unlikely to be achievable. Use of the Beta distribution is briefly mentioned here, but is described more fully later in Section 2.9.

2.5.2 ACTIVITY TIME CALCULATIONS

PERT uses three time estimates to calculate an expected time and variance for each activity:

t_o = an optimistic time (sometimes referred to as the 'crash' job completion time);

t_m = most likely time (or normal duration);

t_p = a pessimistic time (or the decrease in cost per unit increase in completion time from the 'crash' duration).

From this, the expected time (or duration) is derived, becoming the Activity time (A_t), which is assumed to be an independent random variable.

A Beta distribution is used to determine the activity times. For two non–negative parameters, α and β the probability density function *(pdf)* is:

$$f(x) = \left\{ \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\,(\beta)} x^{\alpha-1}(1-x)^{\beta-1} \right. \quad \textbf{(2.1)}$$

for other cases where Γ is the Gamma function:

$$\int_0^\infty t^{s-1}\, e^{-t}\, \mathrm{d}t, \text{ for interger}: \Gamma s = (s-1)! \quad \textbf{(2.2)}$$

and using the discrete random variable 'X', expected activity (A) will be:

$$\mathrm{A(X)} = \frac{\alpha}{\alpha + \beta} \quad \textbf{(2.3)}$$

and when time (t) will be:

$$(t) = t_o + (t_p - t_o)\mathrm{X} \quad \textbf{(2.4)}$$

thus, Activity time is:

$$At = t_o + (t_p - t_o)\left(\frac{\alpha}{\alpha + \beta}\right) \quad \textbf{(2.5)}$$

and most likely time (which will also be the Mode of the distribution) will be:

$$t_m = t_o \frac{(\beta - 1) + t_p(\alpha - 1)}{(\alpha + \beta - 2)} \quad \textbf{(2.6)}$$

when setting $\alpha = 3 + \sqrt{2}$ and $\beta = 3 - \sqrt{2}$ and A_t will be:

$$At = \frac{t_o + t_p + (\alpha + \beta - 2)t_m}{(\alpha + \beta)} \quad \textbf{(2.7)}$$

$$= \frac{t_o + t_p + 4t_m}{6} \quad \textbf{(2.8)}$$

Variance (σ^2) can be also determined in a similar manner:

$$\sigma^2(X) = \frac{\alpha\beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)} \quad \textbf{(2.9)}$$

using Equations 2.4 and 2.8:

$$\sigma^2(X) = (t_p - t_o)^2 \frac{\alpha\beta}{(\alpha + \beta)^2 (\alpha + \beta + 1)} \quad \textbf{(2.10)}$$

and setting α and β as for Equation 2.7:

$$\sigma^2 = \left(\frac{t_p - t_o}{6}\right)^2 \quad \textbf{(2.11)}$$

$$= \tfrac{1}{36}(t_p - t_o)^2 \quad \textbf{(2.12)}$$

An activity is the work completed between two events, in this example event i and j:

$$i \rightarrow j$$

where:

i = Tail event (start of activity);

j = Head event (completion of activity).

Each event can have two possible start and completion times:

i_E = Earliest time of preceding event;

i_L = Latest time of preceding event;

j_E = Earliest time of succeeding event;

j_L = Latest time of succeeding event.

Earliest times are calculated by adding activity times to the earliest completion time of the previous activity (the latest earliest completion time, if more than one activity has to be considered) at any one event node. Latest times are derived by backward calculations subtracting from the earliest network/project finish time determined by the CP.

Assuming that for project *P*, each activity start is determined by i_E, j_E can be calculated by:

$$j_E = \begin{cases} 0, & j = 0 \\ \max\limits_{(i,j) \in P} (i_E + At), & 1 \leq j \leq n \end{cases} \tag{2.13}$$

when n is the final event (a project having $n + 1$ events—0 being the first).

Likewise for project *P*, each activity i_L can be determined. By letting the *P* completion time be denoted by λ:

$$i_L = \begin{cases} \lambda, & i = 0 \\ \min\limits_{(i,j) \in P} (j_L - At), & 0 \leq i \leq n \end{cases} \tag{2.14}$$

when $1 \geq t_n^{(0)}$ i.e., the first activity time.

Time to complete any one activity (A_t) can now use the notation:

$$A_t = i - j \tag{2.15}$$

The following can now be determined:

For start times:

Earliest $\rightarrow i_E$

Latest $\rightarrow j_L - A_t$

For completion times:

Earliest $\rightarrow i_E + A_t$

Latest $\rightarrow j_L$

From this, total, free and independent floats are derived:

Total float (TF)

The amount of time an activity may be delayed or extended without affecting the final completion date of the project i.e., the determination of an activity being critical or not:

$$TF = (j_L - i_E) - A_t \quad \textbf{(2.16)}$$

Free float (FF)

The amount of time an activity may be delayed without affecting any other in the network:

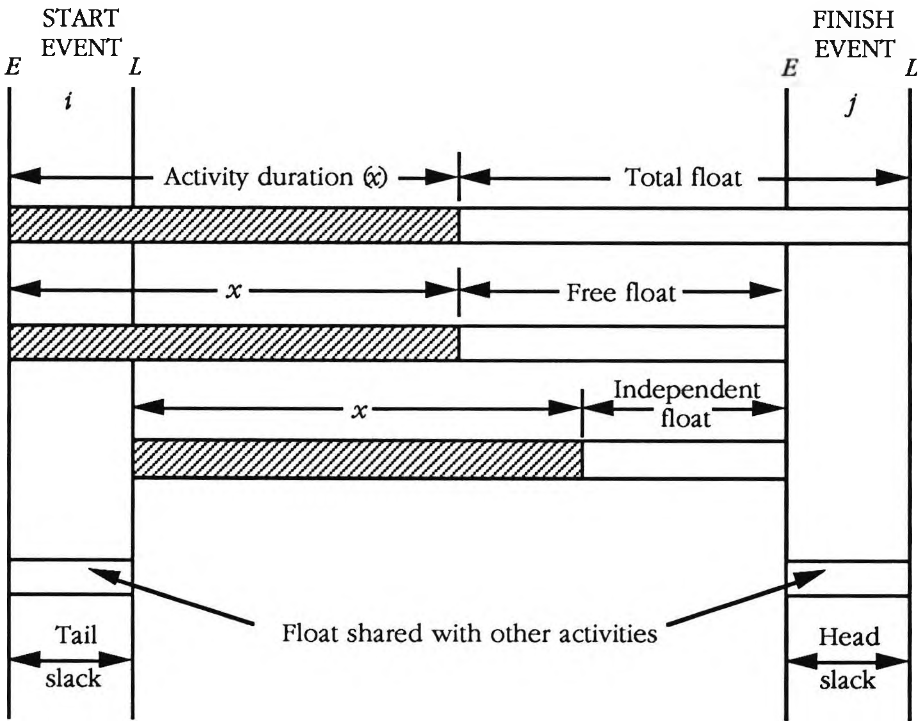
$$FF = (j_E - i_E) - A_t \quad \textbf{(2.17)}$$

Which can also be written as:

$$FF = TF - (j_L - j_E) \quad \textbf{(2.18)}$$

Independent float (IF)

The amount of time an activity can be delayed or extended without



Types of float and event slacks
Figure 2.2

preceding or succeeding being delayed.

$$\text{IF} = (j_E - i_L) - A_t \tag{2.19}$$

In addition to this, there can be slack in any one activity at either the start or finish events. This is the difference between earliest and latest event times or the maximum time to complete any one activity. Thus, for i or start event:

$$\text{Slack } i = i_L - i_E \tag{2.20}$$

Figure 2.2 summarises the various types of float and slacks.

2.6 Planning Process of Starting PERT on a Computer

During the planning process of a LSP, making use of an $i \rightarrow j$, 'activity–on–arrow' PERT network, the following procedure is generally followed, in this example, for a major submarine refuel and refit, using an *Internet 80S* PERT network on a PRIME 750 mini–computer (Ministry of Defence, 1984):

1) The computer is fed with start and completion dates only;

2) computer schedules all activities between these dates, giving earliest and latest scheduled dates;

3) the CP(s) is identified. To decrease time in hand, CPA is undertaken. This involves a detailed inspection of the activity sequence to find areas where time savings can be achieved by either reducing activity cycle time or by paralleling activities previously scheduled in series;

4) having programmed the computer and obtained earliest and latest start and finish dates for every activity, the next stage is to resource load the activities—resource analysis (RA). Having entered resource requirements for each activity and setting resources likely to be allocated during the project, the computer completes the RA and then reschedules activities, ensuring work remains contained within original start and completion dates set for time analysis (TA). The result of RA on the project is to decrease float or amount of slack for each activity, since resource loading places further constraints on the ability to fit the work in between start and completion dates;

5) once agreement has been achieved on likely availability of resources, the RA is again run. Final results of this will become the planned start and finish dates for each activity. CP(s) will also now be identified. Graphs, Bar and/or Gantt charts are produced, together with network and sub–net drawings.

RA scenarios are derived from the resource stacking, resource levelling system developed with earlier versions of PERT (Burgess and Killibrew, 1962;

Davis, 1966; Davis and Heidorn, 1971; Johnson, 1967; Wiest, 1964; Woodworth and Willie, 1975). The system used for this example does not draw resource stack diagrams.

2.7 TIME BASED NETWORKS

These are networks where activities (on link/arrow) represent time (activities are estimated as independent units of work to be undertaken and considered as a separate identity i.e., without consideration being given to preceding or succeeding activities). The ensuing network clarifies to the user activities being worked on. This however would only be representative on a CP, as other activities are on paths where there is float and this system would convey no more information than a conventional network (where activity arrow lengths are not representative of time): '...particularly worthwhile where it is done as an aid to planning and not intended for use after the project is underway.' (Moder *et al.*, 1983: 173). See Figure 2.3 where the ***nuclear system*** is a CP.

Complete networks do not take into account external influences, for example strikes, shortages of resources or "misappropriation" of LRs by other projects being undertaken by the parent organisation. A finalised network prior to project start day can identify LR deficiencies and the logic adjusted to take this into account. Resource levelling problems arise after the start day, when float has been "spent" (see Section 1.5) and several paths of the network converge (near project completion or completion of a path up to a dummy) and many activities need to be completed by the same LRs. This is especially important in LSPs where many activities are stochastic and the original network conceived and drawn without accurate information.

2.8 RESOURCE LEVELLING SCHEDULING HEURISTICS

2.8.1 INTRODUCTION

When trying to plan for the most efficient utilisation of labour resources PERT should carry out a resource levelling exercise. PERT used in PROMISS uses one developed by Woodworth and Willie (1975). Such an algorithm appears to be

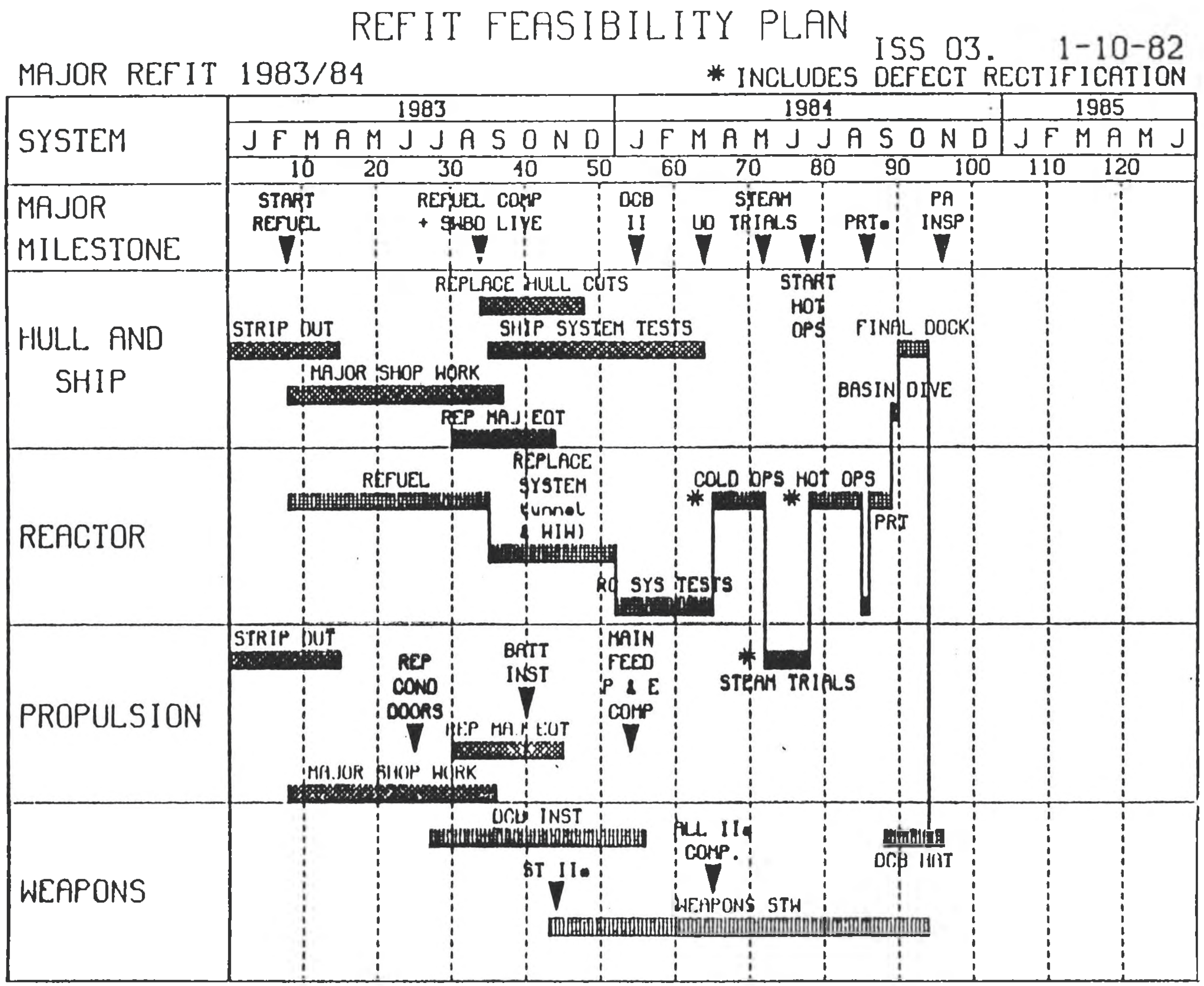


Figure 2.3
Example of a LSP time based network

reliable and accurate enough to provide "a near optimal labour resource profile."

A resource levelling exercise's objective is to flatten out the initial resource profile determined when setting all the network activities at their early start times. This may be the "best" profile to ensure all work is completed at the earliest possible time, but not necessarily for the most "efficient" utilisation of labour resources. Completion of work at earliest start times, requires that labour is provided as and when required. Most projects are invariably manned by a set (allocated) team(s) size and project manager(s) have to make best use of them. By maintaining the initial "early start time plan" the labour resource profile will probably contain many peaks and troughs. Peaks show overloads and troughs underloads of labour which are both inefficient. Resource smoothing attempts to move the profile peaks into troughs, the "perfect" profile being level, implying all labour resources are fully utilised, at a constant rate, for the duration of the project.

Project length is determined by the CP(s) and in most projects, this path(s) only represents a small number of activities as most activities are completed in parallel and therefore have an inbuilt float. This float enables the project to maintain the planned ECD, whilst allowing most other activities to be started at times other than at their earliest start time. Scheduling heuristics make inbuilt use of this tolerance in start and finish times.

There are two constraints used for determining the final labour profile, one based on time and the other on resources. A time constrained scheduling exercise is where various activity start times are fixed, due, for example, to specialist resources being required on other projects. If labour resources are used as the criterion for smoothing, a maximum level available is set and the resource levelling exercise attempts to make best use of them. By shifting the start time of one or all activities away from original (earliest) start times, it is possible to reduce total resource requirement(s) for the whole project (it is also possible that the original activity early start time will provide the best profile). Two common scheduling heuristics (defined as a "rule–of–thumb" i.e., an algorithm for resolving problems) used in the problem of resource constraints are either serial or parallel scheduling (Kelley, 1963) and these briefly detailed below.

2.8.2 SERIAL SCHEDULING

Activities are prioritised and known LR requirements allocated against each one. This heuristic considers LRAs for the project and then ranks each activity in a serial format, taking into account the priority in which the activity has to be completed and LRs available. For example, if five labour units are required for any one activity, then this will not be started until sufficient LRs are released from a completed activity. Early start times are moved to the right in the network until sufficient LRs become available.

2.8.3 PARALLEL SCHEDULING

Activities are prioritised as for serial scheduling, but in this case grouped into start times. Allocation of LRs are made in priority order at each time period. If an activity cannot be started it is moved to the top of the succeeding period group. This method is most commonly used, even though computation time is higher than in serial scheduling.

2.8.4 PRACTICAL ASPECT OF SCHEDULING HEURISTICS

These scheduling schemes do not take into account the practical aspect of finding enough LRs when work has to be carried out. What is the likelihood of having the required number of LRs when the planned *latest* activity start time (i_L) arrives? Conversely, these systems tie the hands of LMs who may have slack LRs available. He may be able to start a later scheduled activity with fewer LRs then required to complete the whole activity as scheduled or planned, but is prevented from doing so.

2.9 BETA DISTRIBUTION FOR DETERMINATION OF PERT ACTIVITY TIMES

As PERT is used as a primary model within PROMISS, consideration is given to the accuracy of activity times used (Clarke, 1961a; 1962; Golenko–Ginzburg, 1989; Grubbs, 1962; Littlefield and Randolph, 1987; Sasieni, 1989). Reliability of this data must be within bounds of a reasonable confidence level, as it will also be source data for the integrated mathematical simulation model. Validity of the simulation model, from which the PM makes his initial decisions, will be enhanced if both the model structure (Forrester, 1961; Meadows *et al.*, 1972) and data are representative of the RWS of Interest.

MacCrimmon and Ryavec (1964) suggested that a Beta distribution be used in determining activity times. They considered errors in these activities as being caused by: 1) distribution assumptions; 2) method of estimating the standard deviation (by up to 30%) and mean optimistically; and, 3) inexactness of time estimates. Problems are exacerbated when there are a number of near–critical paths. Lukaszewicz (1965) made a correction to MacCrimmon and Ryavec's assumptions concerning a symmetrical Beta distribution. It was shown that greatest errors occur when the activity time distribution is a quasi–uniform distribution. Farnum and Stanton (1987) however, consider the use of the Beta distribution (rather than a Normal distribution) for providing improved accuracy when activity estimates are near upper and lower bounds of a distribution curve.

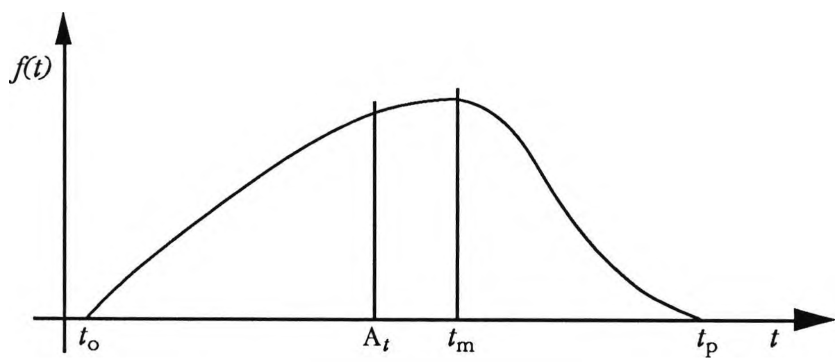
As previously described (Section 2.5.2) there are three possible time estimates for any one activity; t_o, t_m and t_p.

Two most commonly used means of deriving A_t are: $(t_o + t_p)/2$; and also Equation 2.8 being: $(t_o + t_p + 4t_m)/6$. When considering the Beta distribution (derived from Grubbs, 1962):

$$f(t) = \left[\frac{1}{(t_p - t_o)^{\alpha + \beta + 1}\,\beta(\alpha + 1,\ \beta + 1)}\right](t - t_o)^{\alpha}\,(t_p - t)^{\beta}, \quad \textbf{(2.21)}$$

when: $(t_o < t < t_p)$, otherwise: $f(t) = 0$

The subsequent graph, Figure 2.4, for $f(t)$ approximates:



Beta distribution curve of $f(t)$ in Equation 2.18
showing activity times t_o, t_m and t_p with actual time of completion
Figure 2.4

The general form of the density function will be:

$$f(t) = \frac{\Gamma(\alpha + \beta)\,(t - t_o)^{\alpha - 1}(t_p - t)^{\beta - 1}}{\Gamma(\alpha)\,\Gamma(\beta)\,(t_p - t_o)^{\alpha + \beta - 1}}, \quad \textbf{(2.22)}$$

when: $t_o < t < t_p$, with α and $\beta > 0$

When activity times for t_o and t_p are known or estimated, a linear transformation can be used:

$$X = \frac{Y - t_o}{t_p - t_o} \tag{2.23}$$

giving a standardised distribution form:

$$f_X(t) = \frac{\Gamma(\alpha + \beta)}{\Gamma(x)\,\Gamma(\beta)} x^{\alpha - 1}(1 - x)^{\beta - 1}, \tag{2.24}$$

when: $0 < t < 1$, with α and $\beta > 0$.

From Equation 2.21, the Mean (average) value of t, $E(t)$, will be:

$$E(t) = \frac{\alpha + 1}{\alpha + \beta + 2} \tag{2.25}$$

which is transformed into:

$$E(t) = t_o + (t_p - t_o)\frac{\alpha - 1}{(\alpha + \beta + 2)} \tag{2.26}$$

Variance (mean of squared deviations) of t, σ_t^2, will be:

$$\sigma_t^2 = \frac{(\alpha + 1)(\beta + 1)}{(\alpha + \beta + 3)(\alpha + \beta + 2)^2} \tag{2.27}$$

which can be transformed into:

$$\sigma_t^2 = \frac{(t_p - t_o)^2(\alpha + 1)(\beta + 1)}{(\alpha + \beta + 3)(\alpha + \beta + 2)^2} \tag{2.28}$$

Mode (greatest frequency), in this case the most likely time of t, t_m, is found by setting $f'(t) = 0$ to become:

$$t_m = \frac{t_o(\beta - 1) + t_p(\alpha - 1)}{(\alpha + \beta - 2)} \tag{2.29}$$

Variance σ_t^2, will be:

$$\sigma_t^2 = \tfrac{1}{36}(t_p - t_o)^2 \tag{2.30}$$

The above equations are for individual, rather than average random values of 't'. It must be noted when dealing with stochastic time estimates commonly found in LSPs, values of t_o, t_m and t_p are subjective. Criticism can therefore be made of the distribution, if values of t_o and t_p are extreme i.e., positioned too far out in the distribution curve tails. As Grubbs (1962) pointed out, each of the values for *'t'* have their own Variance i.e., $\sigma_{t_o}{}^2$, $\sigma_{t_m}{}^2$ and $\sigma_{t_p}{}^2$, highlighting the need for *objective*, rather than *subjective*, estimates for work undertaken to complete each activity. Within industry, there is an extensive data bank for time estimators to accurately determine times required to complete work (British Standard Times), thereby reducing such uncertainty.

The Modal value of the Beta distribution, the t_m, is obtained by using the time estimators' subjective knowledge (experience). This value is converted to an estimate of the mean μ_t:

The most likely Mean, $\hat{\mu}_t$, for the most likely Modal value, $\hat{t}_m$:

$$\hat{\mu}_t = \frac{4\hat{t}_m + 1}{6} \tag{2.31}$$

and by making assumptions that the Standard Deviation (the square root of the variance—as a measure of dispersion of all values) of t, σ_t, is approximately one–sixth of the range:

$$\hat{\sigma}_t = \tfrac{1}{6} \tag{2.32}$$

Together, with a wide range of values of t_m, the σ_t will not be significantly affected by values of α and β; $\sigma_t(\alpha, \beta) \equiv \sigma_t(\alpha - 1, \beta - 2)$, and when considering the assumption made in Equation 2.32, the following is obtained:

$$\sigma_t(\alpha - 1, \beta - 1) \equiv \sigma_t(\alpha, \beta) \equiv \tfrac{1}{6} \tag{2.33}$$

As $(\alpha - 1)$ and $(\beta - 1)$ are parameters of the distribution, its Mean will also be the same as the Mode, for one with parameters α and β, therefore:

$$\mu_t(\alpha - 1, \beta - 1) = t_m(\alpha, \beta) \tag{2.34}$$

when: α and $\beta > 2$
thereby giving the parameter values for α and β as:

$$\alpha = \left[\frac{\mu_t(1 - \mu_t)}{\sigma_t^2} - 1\right]\mu_t \tag{2.35}$$

$$\beta = \left[\frac{\mu_t(1 - \mu_t)}{\sigma_t^2}\right](1 - \mu_t) \tag{2.36}$$

thus, when considering Equation 2.34, the Equations 2.35 and 2.36 will follow the format:

$$\alpha - 1 = \left[\frac{\mu_t(1 - \mu_t)}{\sigma_t^2(\alpha - 1, \beta - 1)} - 1 \right] \mu_t \qquad (2.37)$$

$$\beta - 1 = \left[\frac{\mu_t(1 - \mu_t)}{\sigma_t^2(\alpha - 1, \beta - 1)} \right] (1 - \mu_t) \qquad (2.38)$$

Considering Equation 2.33; Equations 2.37 and 2.38 can be solved for α and β, then substituted into Equation 2.23 i.e:

$$\mu_t = \mu_t(\alpha, \beta) = \frac{\alpha}{\alpha + \beta} \qquad (2.39)$$

Mean (or expected activity time) will be:

$$\mu_t = \frac{(36t_m^2 + 1)(1 - t_m)}{(36t_m + 1)(1 - t_m) + t_m} \qquad (2.40)$$

and utilising Figure 2.5 (Farnum and Stanton, 1987: 289), i.e., using:

$$\mu_t = \frac{(36t_m^2 + 1)(1 - t_m)}{(36t_m + 1)(1 - t_m) + t_m}, \text{---} \ \mu_t = \frac{4t_m + 1}{6} \qquad (2.41)$$

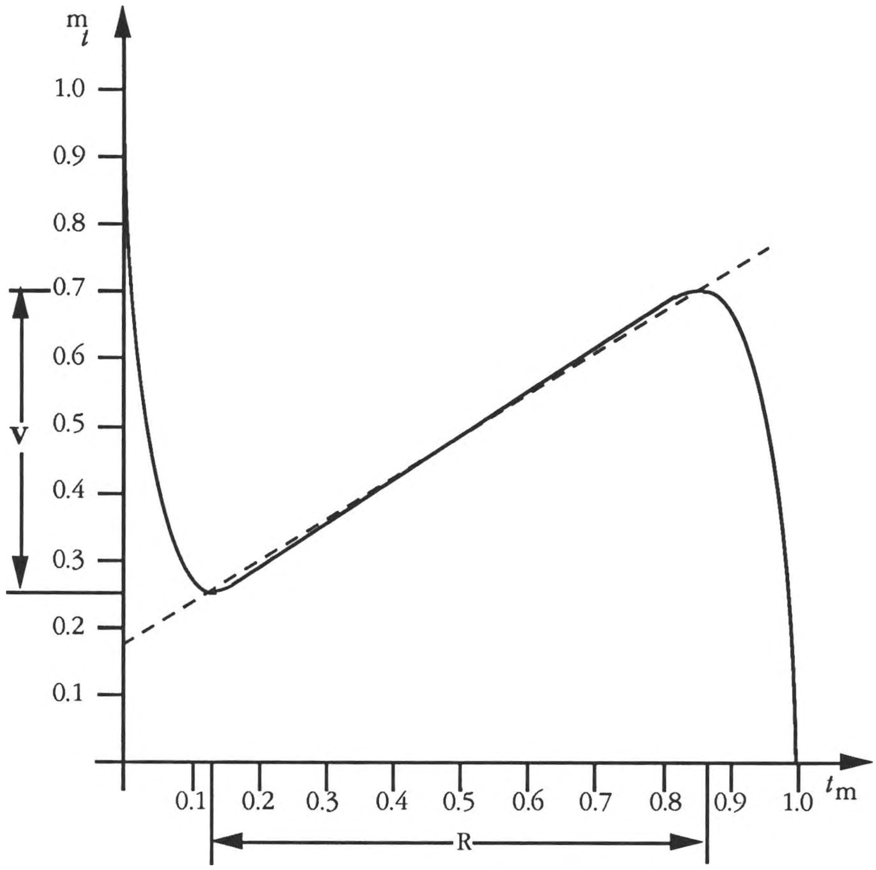
It is now seen that results from Equation 2.31, ('V' in Figure 2.5) are within 0.02 of those from Equation 2.40 if μ_t is between 0.13 and 0.87 (the range 'R'). Modes for t_o and t_p outside the values of 'R' occur when t_m is closer to either of the extreme values of t_o and t_p, when for example t_p is a large "guesstimate" rather than a conservative estimate, thereby causing the Standard Deviation to be < 1/6.

To overcome problems of poor estimates of α and β affecting the overall distribution, it is recommended that the following are used to improve estimates for m_t and s_t outside range 'R' in Figure 2.5.

* If $0.13 \le \hat{t}_m \le 0.87$,

$$\hat{\mu}_t = \frac{4\hat{t}_m + 1}{6} \qquad (2.42)$$

$$\hat{\sigma}_t = \frac{1}{6} \qquad (2.43)$$



Graph of Equations 2.30 and 2.39 of Beta distribution (Estimates of Mean)
Figure 2.5

as per Equations 2.31 and 2.32, but if:

* $\hat{t}_m < 0.13$,

$$\hat{\mu}_t = \frac{2}{(2 + \frac{1}{\hat{t}_m})} \quad \textbf{(2.44)}$$

$$\hat{\sigma}_t = \left[\frac{\hat{t}_m^2 (1 - \hat{t}_m)}{(1 - \hat{t}_m)} \right]^{\frac{1}{2}} \quad \textbf{(2.45)}$$

and if:

* $\hat{t}_m > 0.87$,

$$\hat{\mu}_t = \frac{1}{(3 - 2\hat{t}_m)} \quad \textbf{(2.46)}$$

$$\hat{\sigma}_t = \left[\frac{\hat{t}_m (1 - \hat{t}_m)^2}{(2 - \hat{t}_m)} \right]^{\frac{1}{2}} \quad \textbf{(2.47)}$$

Taking into account the Beta distribution and problems of poor estimation of work rather than ignoring it, would help to enhance the accuracy of PERT data. The significance of stochastic activity times affecting final Project Completion Date (PCD) of a LSP has to be acknowledged. However, to what extent poor estimates outside the accepted t_m range of $0.13 \leq t_m \leq 0.87$ affects the final completion date, is debatable. PMs (or project staff), as LSP decision maker(s), must be aware of spurious estimates of activities; but the overall effect that this would have (initially only significant) on the CP(s). Other activities would have the advantage of slack and float within their respective paths (between restraints/dummies) and can thus absorb, to a certain extent, time estimate inaccuracies—provided utilisation of this tolerance is controlled. Such inaccuracies would have an effect later on during the completion of project, should previous non–critical paths become critical. What ***would*** affect timely completion of the LSP is a poorly maintained and updated PERT (and database) during the project duration.

Complications in determining which equation (2.42, 2.44 or 2.46) to use arise from human error. For example, using an estimated time with Equation 2.44, when Equation 2.46, should be used would be more erroneous and detrimental to LSP completion, than just using Equation 2.8 and letting A_t:

$$= \frac{t_o + t_p + 4t_m}{6} \tag{2.48}$$

Moder and Rodgers (1968) carried out work to find the most accurate method of determining PERT time estimates. They consider that the use of a single time estimate (t_m) gave a biased estimate, unlike a calculation using t_o, t_m and t_p together. Therefore, statistical estimates of t_o and t_p have been added to reduce this bias. "Standard" or "classical" PERT assumes t_o and t_p (the end points of an activity time distribution curve) are 0 and 100 percentiles of the range respectively. Experiments with estimators of differing skills/knowledge have concluded that estimates should be based on 5 and 95 percentiles. This has enabled Equation 2.8 to be rewritten as:

$$\hat{\mu}_t = \frac{t_{o(5)} + t_{p(95)} + 4t_m}{6} \tag{2.49}$$

Variance for estimates would have the following format:

$$\hat{\sigma}_t^2 = \frac{(t_{p(95)} - t_{o(5)})^2}{10.2} \tag{2.50}$$

which has a less negative bias than Equation 2.12.

These equations were tried during the building of the PERT model within PROMISS and the subsequent results produced were identical to those from Equation 2.48. Research and reading failed to provide any responses to this paper and the authors' assumptions. Further analysis is therefore not the subject of this research.

2.10 Simulation Modelling to Determine PERT Critical Path

2.10.1 Introduction

There have been a number of attempts to reduce computer processing time in determining the CP(s), for example Aonuma, 1964; Charnes and Cooper, 1962; Clingen, 1964; Fulkerson, 1962; Hammond, 1971; Hooper, 1986. Cook and Jennings (1979), considered the use of "intelligent simulations" as a means of determining likely completion of a Project. Three heuristics are considered in an attempt to reduce CPU time and increase accuracy of output from a PERT network. The computer simulation heuristics are: 1) Min–Max; 2) Path Deletion; and 3) Dynamic Shut–Off.

2.10.2 Min–Max

This method analyses the network twice, firstly using $A_t = t_o$ and then $A_t = t_p$. The first analysis identifies the optimistic critical path(s) (OCP). A second analysis makes a comparison of all activity paths against the OCP. If, during the $A_t = t_p$ analysis, a path is < OCP, then the computer identifies this as a non–critical path (NCP).

2.10.3 Path Deletion

This is computationally the same as Min–Max simulation, for the first one hundred iterations. Those paths not identified as a CP are removed from the remainder of the simulation and identified as NCPs.

2.10.4 Dynamic Shut–Off

This is similar to Min–Max simulation, but iterations are controlled dynamically. The cumulative frequency distribution *(cfd)* is determined after each one hundred iterations and then compared. Making use of the Kolmogorov–Smirnov (K–S) test, the significant difference of each *cfd* is also compared. If there is no significant difference at the 0.05 level, then the simulation is terminated. The K–S test is briefly described below in Section 2.11.

When dealing with a LSP, deletion of paths in the latter two simulation methods could bias any subsequent results.

2.11 KOLMOGOROV–SMIRNOV TEST

K–S is a non–parametric test, concerned with "goodness of fit" between the distribution of a set of sample values (observed activity times) and a theoretical distribution (estimated activity times). Divergence between the H_0 (Null Hypothesis) distribution and actual distribution H_1 (Alternative Hypothesis) is calculated, so as to determine if the observed activity times would actually occur, if they were a random sample of the H_0 distribution. Below is a brief description of the *one sample* test (Siegel, 1956):

Let $F_0(X)$ = the theoretical *cfd* under H_0 for any value of 'X', the activity times.

Thus:

$F_0(X)$ is the proportion of activity times expected to have times equal to or less than 'X';

Let $S_N(X)$ = the observed *cfd* of the random sample of *'N'* actual activity times.

Where:

'X' is any possible time $S_N(X) - k/N$, where *'k'* is equal to the number of observations equal to or less than 'X'.

The K–S test is only concerned with maximum deviation 'D', between $S_N(X)$ and $F_0(X)$:

$$D = \text{maximum}[(F)(X) - S_N(X)] \qquad \textbf{(2.51)}$$

From a table of Critical Values of 'D' (those dependent upon *'N'*) in the K–S test, significant difference can be determined.

This test treats each observation individually and does not lose information by combining groups of activity times, as one would when using the Chi–square test, (Massey, 1951; Scharank and Holt, 1967; Siegel, 1956).

2.12 CONCLUSION

PERT network schedules are now ubiquitous with projects of any size. For the last thirty years the fundamental philosophy, format and use have remained the same. Throughout the proliferation of PERT software the same mathematical foundation described and detailed in this chapter is maintained (although few make use of the Beta distribution to determine activity times). In general terms, only differences in user interfaces, output information format and size of project (resources etc.) differentiates various software packages now available, see Appendix 1.

PERT remains an excellent logic work guide, but recent developments in computer technology (both hardware and software) have had little radical impact on its use, unlike other computerised planning systems. The limitation of PERT are founded on its poor performance as a decision making tool in projects of any great size or complexity. Many decision making systems in other working environments now utilise mathematical based simulation models. Such models can be more efficient in computer time and used to provide specific information, taking into account the whole situation.

This chapter considers the mathematical and statistical foundations of PERT. Information from this review is used as the foundation of such a system, forming the base program for PROMISS. The next chapter looks at how a PERT computer program is developed.

CHAPTER 3

Development of a Logic Based Model

3.1 INTRODUCTION

Most basic PERT programs are based upon a very simple sub–routine. Examples of these are shown in Firth, 1983 and Thierauf *et al*, 1985, written in BASIC. Both these programs provide insight into how to develop a PERT that could be used for a project network of up to the two hundred activities needed for this research. These two examples in their original interpreted execution format, are too restrictive in how they use or occupy computer memory. What is needed for a DSS (PROMISS) is a program that will execute commands as quickly as possible. Interpreted rather than compiled programs tend to be slow to execute. Efficiency of interpreted program execution rate is entirely dependent upon the way it is written. In general terms programs written in BASIC operate at a high language level and execution is as a result of interpretation of each line of code (a number of versions of BASIC are now compiled rather than interpreted) rather than being compiled to a lower level language. By using a programming language that can be compiled, source code can be protected from copying by providing users with compiled versions (.EXE versions).

3.2 PROMISS—WHY TURBO PASCAL?®

To make the program run as quickly as possible a decision was made to write PROMISS (made up of two executable programs connected together by a third root program) in Turbo Pascal initially in Version 4.0 and then 5.5 (no use was made of Object Oriented Programming (OOPs) options available with this later version). Version 5.5 provided new features such as debugging windows and variable tracing, essential for large programs. Users are unaware that PROMISS is three combined executable programs as it appears to be seamless. The reasons for using Turbo Pascal are:

- It is a well structured high level language and easily understood;

- a program can be broken down into unit files making it easier to debug, modify and expand;
- it is a fast compiler (to memory and/or disc), making it easier to implement changes;
- executions are quicker because it compiles before being run;
- there is a high degree of memory management available;
- future development can take place by making use of overlays and/or memory extenders therefore not restricting size of final program after further development.

3.3 PERT PROGRAM

Below is a description of how core and significant parts of the program code are written.

Fundamental structures are initially defined as:

IArray : an integer array
QArray : two dimensional integer array
XArray : floating point array

Initial variables are set to zero, thus ensuring there are no residue values from previous calculations or iterations.

After initialising variables the user is asked for time units to be displayed. Time units (Hours, Days and Weeks) provide sufficient information for most users. Calculations used in all parts of PERT are in hours but results can also be displayed in the other formats.

```
GetReply('Time unit to use: (H)our (D)ay or (W)eek?',['H',
'D','W'],TRUE,T_UnitType);
GetTimeUnit(T_UnitType, HoursPerDay, HoursPerWeek, T_Unit,
T_UnitName);
```

(Ref. File PERT.PAS 133–134)

The default is set at eight hours per day and forty hours (five days) per week. As this is a development program it is unnecessary to provide a diary function available with some commercial programs. Once a data file has been selected and obtained, core data can be displayed.

An essential sub–routine in PERT is the means of sorting activities into preceding and succeeding order. One means of doing this is by use of a Q–Matrix i.e., matrix Q (of type QArray). A two dimensional matrix is set up (in

PROMISS to handle two hundred activities). A Q–Matrix follows the format of Q [i, j] = x, where x = 1 if activity i precedes activity j, otherwise 0.

Results from this analysis are stored in an array X (of type IArray), containing sorted activities in descending order, i.e., activities with no preceding activities are placed at the top and so on. This array is built by initially making two lists: FList and PList. FList[i] contains 1 if activity being sorted is already in order. Should an activity be succeeded by another it is then stored in the PList[i]. FList and PList are ordered through a function, see below:

```
Function Sorted(Q : QArray; F : IArray; np : Integer) :
Boolean;
var m : integer;
boo : array[1..MaxCount] of boolean;
bee : boolean;;

begin
  for m := 1 to np do
    If ((Q[X[i],m] = 0) or (FList[Q[X[i],m]] = 1)) then
    boo[m] := true
  else
    boo[m] := false;
    bee := true;
  for m := 1 to np do
    bee := bee and boo[m];
    sorted := bee;
end;
```

(Ref. File PERT.PAS 147–164)

Ordering starts by checking if an activity i needs ordering, i.e., does it have a preceding activity or not (a relative boolean value of 1 or 0). If a question relating to the Q–Matrix is 0 and FList [i] = 1, then a boolean value 1 is assigned so that comparison can be made. All other comparisons providing a false answer have a 0 value. Once 1 and 0 values are known they are then sorted.

Sorting activities to identify start and finish order is carried out by the following routines:

For identifying starting activity(s).

```
for i := 1 to np do Q[0,i] := 0;
```

(Ref. File PERT.PAS 171)

For identifying finishing activity(s).

```
for i := 1 to np do Q[No_Recs+1,i] := 0;
```

(Ref. File PERT.PAS 172)

It is essential that preceding activities are identified or marked (flagged) on the PList. This is achieved by the following :

```
    for j := 1 to np do
  If (Q[i,j] <> 0) then Plist[Q[i,j]] := 1;
```

(Ref. File PERT.PAS 177–178)

Note: array had been initialised to zeros at beginning of routine.

If an activity precedes no other, it must therefore be a finishing activity (one with no succeeding activity already identified i.e., number of records + 1). It is also necessary, in the case of PROMISS, that no more than two hundred activities are entered, so *j* must not exceed this number.

```
for i := 1 to No_Recs do
  If Plist[i] = 0 then
    begin
      j := j + 1;
      If j > MaxCount then
        begin
        FlushKeyBuffer;
        Prompt('Number of ending activities exceeding
        maximum (MaxCount).',True);
        Abort('');
    end
```

(Ref. File PERT.PAS 180–191)

As each activity in PROMISS's PERT is also limited to five preceding activities per event (due to complications in displaying any more) it is necessary that the Q–Matrix and sort procedure is aware of this limitation. It needs to check if: a) this number is not exceeded; and b) if there are up to five, they are all identified. This prevents, on finding one preceding activity, the search stopping at that point and sorting out the next activity, thus ignoring other associated preceding activities.

Once search and ordering for each activity is complete it is flagged (preventing unnecessary further searches).

```
i := 1;
repeat
  for j := 1 to 5 do
  begin
    If (Q[i,1] = 0) and (Q[i,2] = 0) and
      (Q[i,3] = 0) and (Q[i,4] = 0) and (Q[i,5] = 0) then
        No_Init := False;
  end;
  i := i + 1;
until (NOT(No_Init) or (i > No_Recs));
```

(Ref. File PERT.PAS 193–202)

```
FList[x[i]] := 1;
```

(Ref. File PERT.PAS 213)

which is then used to put activities into correct order. This facility is required as users can enter activities in any order, adding and subtracting as they change plans. Once activities have been ordered they are put into an array X (of type XArray) which is a temporary array.

```
temp := x[i];
x[i] :=  x[p0];
x[p0] := temp;
p0 := p0 + 1;
```

(Ref. File PERT.PAS 214–217)

If, as sometimes happens, a user incorrectly enters an activity that precedes, instead of succeeds, another activity, an illogical loop will occur. For example if the sequence of activities should follow:

1—8—99—2—7—4

and the user has entered activity 8 as being preceded by activity 7 then a logic error has occurred (activities are interdependent).

```
    else j := j + 1
end;
end; (* while *)
If Not(j <= (No_Recs*No_Recs)) then
  Abort('Activities  interdependent. Aborting . .');
```

(Ref. File PERT.PAS 219–223)

When above sorting routine has been completed, each activity will be in an order such that Earliest and Latest Start and Finish Times can be calculated. Once Earliest Start and Finish Times are identified CP(s), their length and activities, will be flagged (the final table will identify, to a user, critical activities with a "Cr" flag).

Calculations are undertaken and stored in four separate arrays:

EST[i] Early Start Array
EFT[i] Early Finish Array
LST[i] Late Start Array
LFT[i] Late Finish Array

The following routine calculates Expected Time (Ex) and Variance (Vr).

```
begin
  Ex  := (Ot + 4 * Lt + Pt) / 6;
  Vr  := (Pt - Ot) / 6;
  Vr  := Vr * Vr;
end;
```

(Ref. File UTILS.PAS 745–749)

The Q–Matrix is analysed, sorted and computes Early Start and Finish Times through all paths in the network. This is started by the following routine:

```
For i := 1 to No_Recs + 1 do
  begin
    For j := 1 to MaxCount do
      begin
        p := 0;
        repeat
          If (Q[i,j] <> X[p]) then
            p := p + 1;
        until Q[i,j] = X[p];
        S := EFT[p];
        If EST[i] < S then EST[i] := S
      end;
    EFT[i] := EST[i] + ExT[i]
  end;
```

(Ref. File PERT8.PAS 518–531)

Early Start Time of a succeeding activity will be equal to the latest Latest Finish Time of all activities preceding that activity i.e., the *i* event.

```
If EST[i] < S then EST[i] := S
```

(Ref. File PERT.PAS 267)

After Early and Finish Times have been computed the network length is determined—the CP(s). Once Earliest Finish Time of the last activity is known, Latest Finish Times can be calculated. Expected Times are subtracted from Latest Finish Time of succeeding activities, working from right to left through the network. Before calculations can start, variable *S* is made a large number (a number that will greatly exceed any likely length of a network i.e., 10×10^7) so computation will continue until the starting activity is reached.

```
S := 1e7;
For p := No_Recs DownTo 1 Do
  begin
    for i := 1 to No_Recs + 1 Do
      begin
        For j := 1 to MaxCount Do
          If Q[i,j] = p then
            If LFT[i] < 1e7 then
              begin
                S := LStT[i];
                If S < LFT[p] then
                LFT[p] := S;
              end;
      end;
```

(Ref. File PERT.PAS 293–307)

If there is no difference between Late and Early Finish Times i.e., zero float, then that activity is on a CP (in practice it is noted that not all activities with zero float will be on the CP(s) as such e.g., activities preceding a fixed milestone). To prevent round–off errors causing a false result, comparison is made with a variable identified as Tol (i.e., equal to 10×10^{-6} a value that is very small).

Activities identified as being on a CP are recorded in the Critical Array Cr[i]:

```
If ((Abs(LFT[i] - EFT[i]) < Tol)) then
  begin
    Cr[i] := TRUE;
    If ((LFT[i] <= LFT[No_Recs+1]+Tol) and (LFT[i] >=
    LFT[No_Recs+1]-Tol)) then
      EndingActivityNo := i;
  end;
```

(Ref. File PERT.PAS 320–325)

3.4 Resource Levelling

3.4.1 Levelling Algorithm

The aim of resource levelling is to reduce the number of activity labour profile troughs and peaks and utilise resources at a continuous rate as possible, for the duration of a project. An alternative is to start each activity at the earliest possible time and use labour resources as and when required. Although this may be the "ideal" means of running a project, it is unlikely to be achievable in many situations. Whether or not a project is running inside a larger organisation or as a separate identity, a pool of labour resources being ready for use, as and when required, is unlikely to be available. The most likely scenario is that an allocation of resources will be made (or decided on) that will cause overload in the work peaks (increased overtime) and underload in the troughs (resources not working, but still being a cost to the project). One such criterion often used is to allocate an average requirement for the given work.

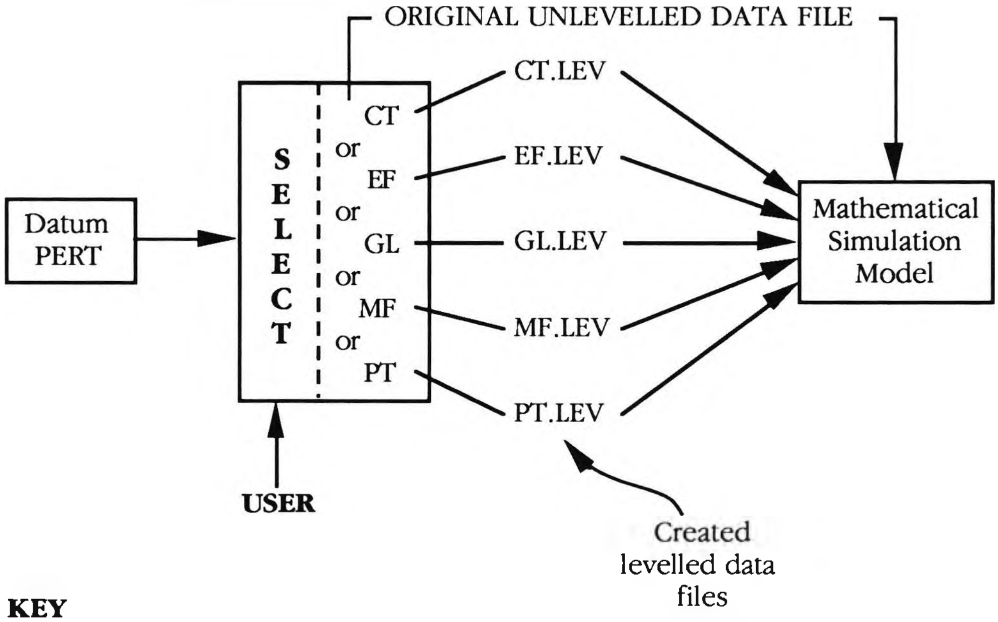
Resource levelling utilises float found in non-critical paths so that activities can be moved to reduce peaks and troughs i.e., "flattening" the labour resource profile. Woodworth and Willie's (1975) levelling/resource algorithm, used in PROMISS's PERT, was originally designed to resolve problems with multi–resource scheduling in multi–projects. This work was a development of that started by Burgess and Killibrew (1962) as a means of measuring alternative positions of activities along non–critical paths. The fundamental principle of these resource levelling algorithms is that of considering each activity in turn and moving it, initially, to the right, one working day (eight hours) at a time.

This was set as a default in the PROMISS program because it is difficult and unlikely to manage movement of resources in smaller time units. Practicalities of moving labour half way through one job, and in the middle of a day, only to shift them back half way through the following day, would result in a considerable amount of unproductive work. Activities, however, do start and finish during the day and LRs are thus moved to the next activity.

Measurement used in shifting activities and considering the best start time (a measure of effectiveness) is a sum–of–squares value. This is determined by adding all currently worked on activities in each day, squaring the quantity of resources used and summing them over project duration, determined by the CP. This levelling exercise considers one priority resource type at a time. PMs will be aware of which resource type may be critical (one that may have problems in supply/allocation for all the work being undertaken during the project) to ensure that project completion date will not slip to the right.

At the beginning of this scenario a PM is asked, by PROMISS, to select a priority resource to be levelled. Before making this selection a resource table (labour consumption profile) can be obtained from the PERT. PMs will have to run separate levelling exercises to determine requirements for each resource (one levelled with priority given to any one resource from five used in PROMISS: Constructive Trades; Electrical Fitters; General Labour; Mechanical Fitters and Painters). These resource files are used to feed labour requirements into the simulation model separately. PMs are also able to select the original, unlevelled, resource table. PMs can therefore consider each and all options with respect to utilisation of LRs. If priority is given to any one LR, data concerning all resources used to complete each activity will also be transferred into the simulation model. This procedure is shown in Figure 3.1 below.

By using a sum–of–squares method one is able to "optimise" resource requirements for the project/network. Optimisation in this case is as close as can be practically (and theoretically) determined. The algorithm for this process, utilising float on activities other than those on a CP (where activities have zero float), is shown in Figure 3.2 below. Each activity is moved initially, in turn, one time unit to the right. Each resource type used in the project can be prioritised in turn thereby giving the PM maximum freedom to consider all possible LRA scenarios. Coding for this procedure is shown below:



KEY

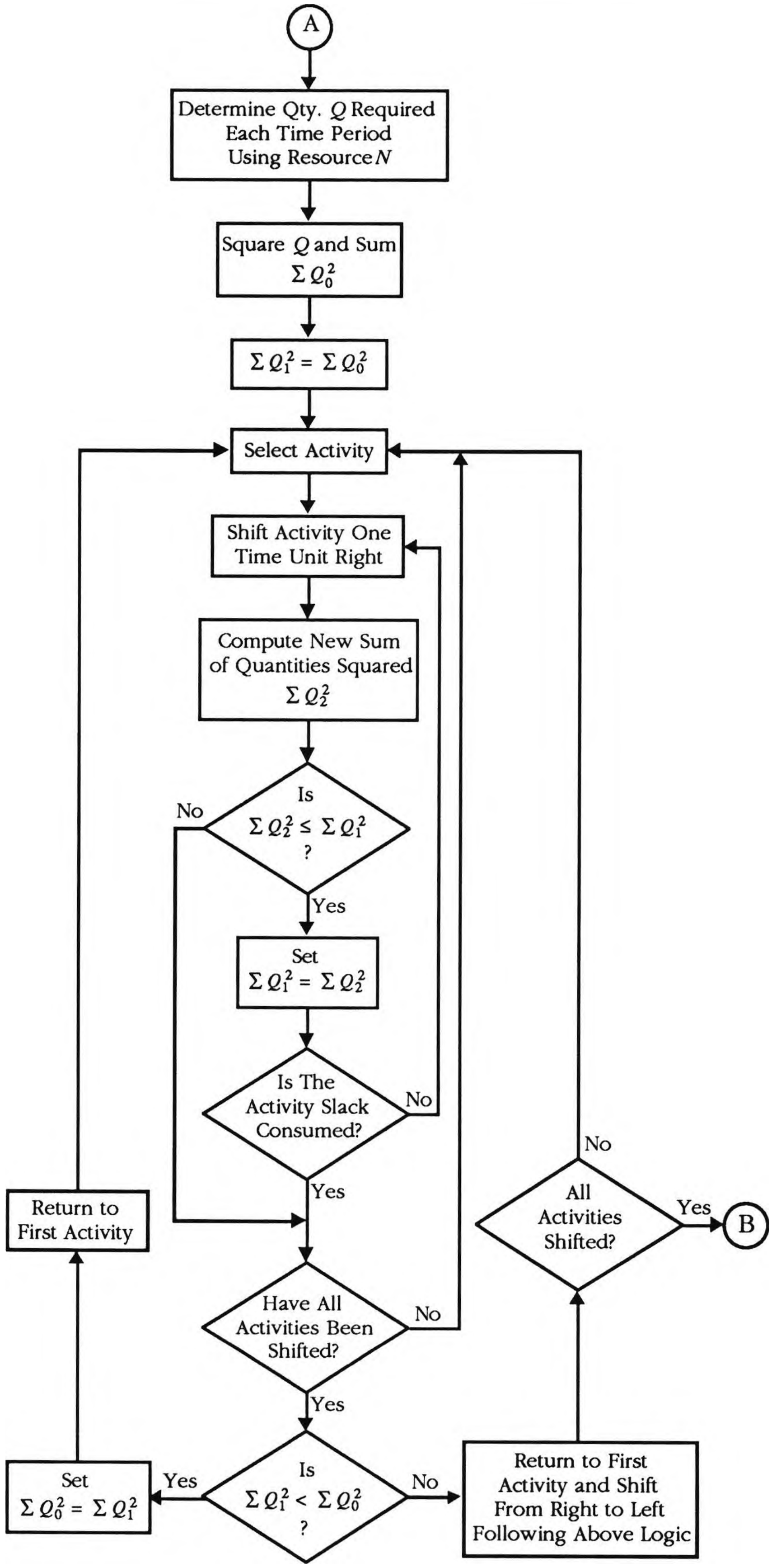
CT Constructive Trades
EF Electrical Fitters
GL General Labour
MF Mechanical Fitters
PT Painters

Transfer of levelled/unlevelled data into the simulation model
Figure 3.1

```
  begin
    SqSum := 0;
    Time := 0;
repeat
      Sum := 0;
      for p := 1 to No_Recs do
        If ((EST[p]+Sh[p]-TOL <= Time) and (EFT[p]+Sh[p]-
        Tol >= Time)) then
          Sum := Sum + Alloc[p,Resource_No];
      SqSum := SqSum + Sum*Sum;
      :
      Time := Time + HoursPerDay;
      :
    until (Time > EFT[No_Recs+1]) or (ExitRequested);
  end;
```

(Ref. File PERT.PAS 373–387)

By starting with an initial high value, (SqSum0 = 10 x 10^7) a comparison between that figure and the one calculated can be made. If the new figure is higher, then the original value is kept. If calculations provide a lower figure, this is kept for comparison with subsequent iterations. Future calculations (the number of iterations determined by the float (in hours) divided by eight, until Latest Finish Time has been reached i.e., Max Shift) are made, between Early Start and Early Finish, until the lowest figure is found.



Resource levelling algorithm
Figure 3.2

```
For i := 1 to No_Recs do
  if (LStT[i] - EST[i]) > Tol then
    MaxSh[i] := Trunc((LStT[i] - EST[i])/HoursPerDay)*
    HoursPerDay;
SqSum0 := 1e7;
:
:
:
If SqSum < SqSum0 then SqSum0 := SqSum;
```

(Ref. File PERT.PAS 606–625)

Calculated, lower values are stored in a shift array Sh[i] for each *i* activity with float (or tail event slack).

```
begin
  j := HoursPerDay;
  repeat
    :
    Temp := Sh[i];
    Sh[i] := j;
    CheckSum;
    :
    If SqSum < (SqSum0-Tol) then
      SqSum0 := SqSum
    else
      Sh[i] := Temp;
    j := j + HoursPerday;
  until (j >= MaxSh[i]);
end;
```

(Ref. File PERT.PAS 629–643)

Once the lowest value is found, its position (recommended start) and value is stored and then the same procedure is carried out for the next activity. This procedure is carried out for all activities with float greater than zero.

```
For i := 1 to No_Recs do
  If ((MaxSh[i] <> 0) and (Alloc[i,Resource_No] <> 0))
  then
    begin
      j := HoursPerDay;
      repeat
        GoToXY(29,5);Write(i:3); GoToXY(43,5);Write(j:5);
        Temp := Sh[i];
        Sh[i] := j;
        CheckSum;
        If ExitRequested then GoTo 0;
        If SqSum < (SqSum0-Tol) then
          begin
            SqSum0 := SqSum
            :
          end
        else
          Sh[i] := Temp;
          :
        j := j + HoursPerday;
      until (j >= MaxSh[i]);
    end;
```

(Ref. File PERT.PAS 623–643)

The problems associated with shifting one activity at a time is its possible effect on other activities—float consumption through the rest of the path(s) affected by that activity. All likely activity start times therefore need to be considered against each other. Combinatorial calculations associated with this are complex and difficult to solve. Burgess and Killebrew (1962) suggested left shifting and again checking for lower sum–of–squares values as a means of resolving this situation. Computation of left shifting is similar to the initial right shifting exercise.

```
For i := 1 to No_Recs do
  If ((MaxSh[i] <> 0) and (Alloc[i,Resource_No] <> 0))
  then
    begin
      j := Sh[i];
      while Not(j < 0) do
        begin
          :
          Temp := Sh[i];
          Sh[i] := j;
          CheckSum;
          If ExitRequested then GoTo 0;
          If SqSum < (SqSum0-Tol) then
            begin
              SqSum0 := SqSum
              :
            end
          else
            Sh[i] := Temp;
            :
          j := j - HoursPerDay;
        end;
      end;
```

(Ref. File PERT.PAS 645–666)

This procedure completes all iterations to pass through the algorithm from point *A* to point *B* in Figure 3.2. Results from temporary Shift Array Sh[i], with the lowest sum–of–squares values (and thus improved resource allocation) are then stored in a permanent file for future analysis, printing and/or use in the simulation model.

3.5 CONCLUSION

This chapter has shown how various aspects of PROMISS's PERT program operates. Strategic parts of the coding, in Turbo Pascal, have been described. Most of the remaining parts of the PERT program are related to either data entry, file management, maintenance or for showing outputs, both on–screen and in hard copy formats.

As PERT is now a widely known concept and many programs written to carry out the same fundamental task of logically sequencing activities, it is considered unnecessary to show the whole computer listing. This would otherwise detract from the fundamental principles of PERT as an integral part of PROMISS and its creation as a DSS for PMs.

CHAPTER 4

Computer Modelling

> Modeling often becomes an end rather than a means. The dedicated modeler reminds one of Pygmalion, the sculptor of Greek mythology. He fashioned a beautiful statue of a girl and fell in love with it. Responding to his plea the goddess Aphrodite brought her to life and he married his model. ... Clearly it seems to be fun for the modelers, but it also is a nightmare for the real world problem solver.
>
> H. A. Linstone,
> *On the Management of Technology: Old and New Perspectives*, 1983

4.1 INTRODUCTION

Simulation modelling, and its derivative computer simulation, originates from physical models made to represent complex real world situations (RWS). It is the process of designing a representative model and conducting experiments (scenarios) for the purpose of either understanding behaviour, or evaluating various strategies (within limits imposed by a criterion or set of criteria) for an operation, (Shannon, 1975). Simulation, by definition, can only include the foreseen and foreseeable, noting that such models are not solved, but run. Use of mathematical based computer simulation models has initially been restricted to those with computer/simulation expertise. This barrier to widespread use is being overcome by use of Artificial Intelligent (AI) interfaces for model building, (Gonzalez and Fernandez, 1986; Khoshnevis and Austin, 1987).

This introduction of reliable digital computers with high–speed processing capacity in the 1950s has made their use for simulation a feasible technique for problem solving (situation analysing). Further advances in hardware technology in the late 1970s and 1980s has further increased processing speeds, enlarged memory and on–line storage capacity, together with a reduction in physical size, cost of purchase and running computers:

> Today the equipment and the technology will allow simulation of anything from man's thoughts to his concept of the universe.
>
> Digital computers are now fast enough to simulate most real systems in real time. And computational results can be made accurate to any desired degree.

> (A critic once observed that even when digital computers' results are wrong, they are precisely wrong!). (McLeod, 1988: 14)

Developments in software design, have enabled simulation techniques to become increasingly used as a decision making tool. Ahuya and Nandakumar, 1985; Gordan, 1977; Law and Kelton, 1982; Maisel, 1976; Rivett, 1972, 1984; Shannon, 1975, with Mayer and Adelsberg, 1985; Zeigler *et al.*, 1979 are amongst a number of researchers who have identified the usefulness of decision making computer simulation techniques.

> The advancement and acceptance of simulation as a method of decision analysis has been directly tied to developments in both computer hardware and software. In the hardware arena, increasing power and decreasing costs have made computers available to organizations of all sizes. (Shannon, 1986: 150)

4.2 Simulation Modelling

Development has progressed since the early 1970s in use of simulation techniques as a foundation of Expert Systems (Edwards, 1982; Flitman and Hurrion, 1987; Halroyd *et al.*, 1985; De Swann Arons, 1983). Expert Systems (ES) have also been used for controlling computer simulations (Doukidis, 1987, with Paul, 1985; Jacucci and Uhrik, 1987; Moser, 1986). This latter point has enabled production of bespoke simulation program generators, from a natural language like dialogue, as a technique for use by an analyst with little computer knowledge, (Doukidis, 1985).

The structure of a simulation system model, irrespective of its complexity, has a mathematical foundation in the form:

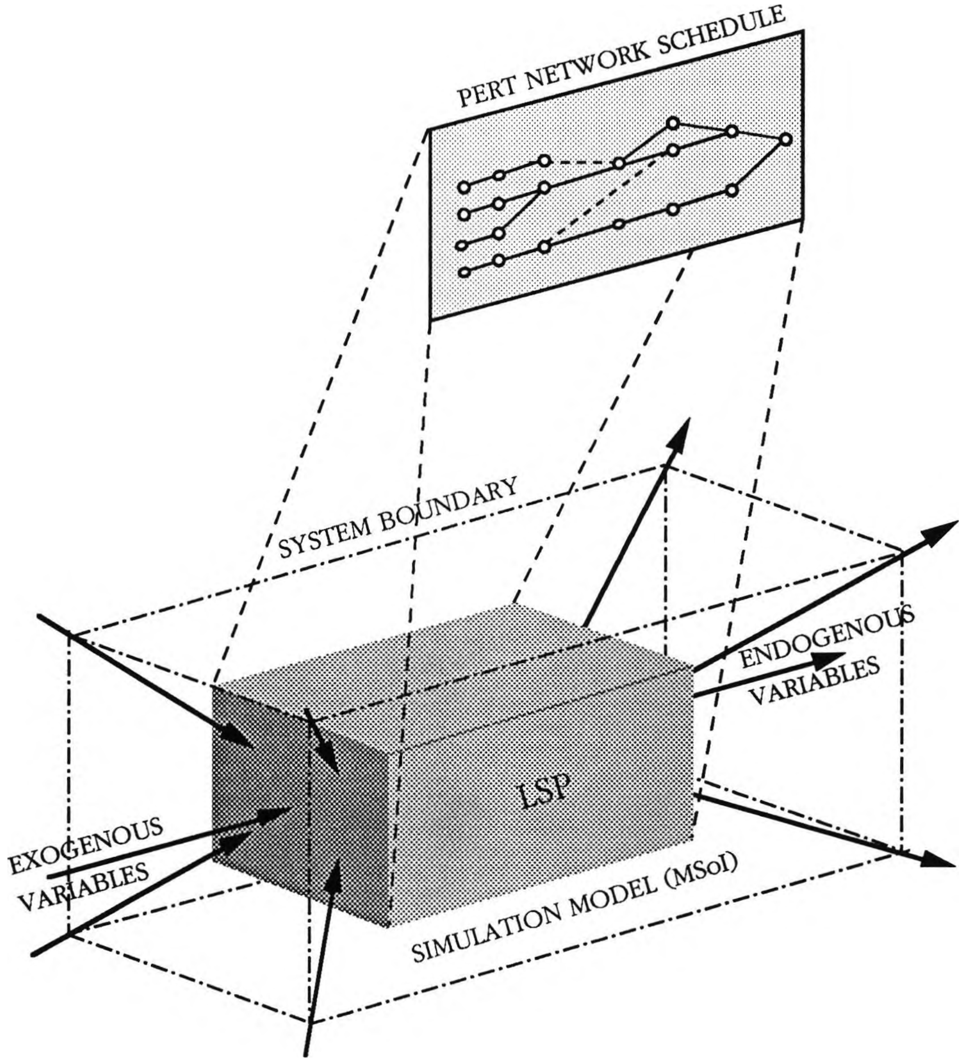
$$\mathbf{E} = f(c, u) \qquad \textbf{(4.1)}$$

where, from the viewpoint of the analyst:

- **E** is the effect of the system's performance;
- c are the controllable variables and parameters;
- u are the uncontrollable variables and parameters;
- f is the functional relationship between c and u which will effect **E**.

The situations considered in this research are open systems. Few human activity systems could ever be considered closed and they will include variables that cannot be controlled. These may be either classified as exogenous or endogenous. When looking at any System of Interest (SoI) inputs from the environment are exogenous, whilst those produced by the

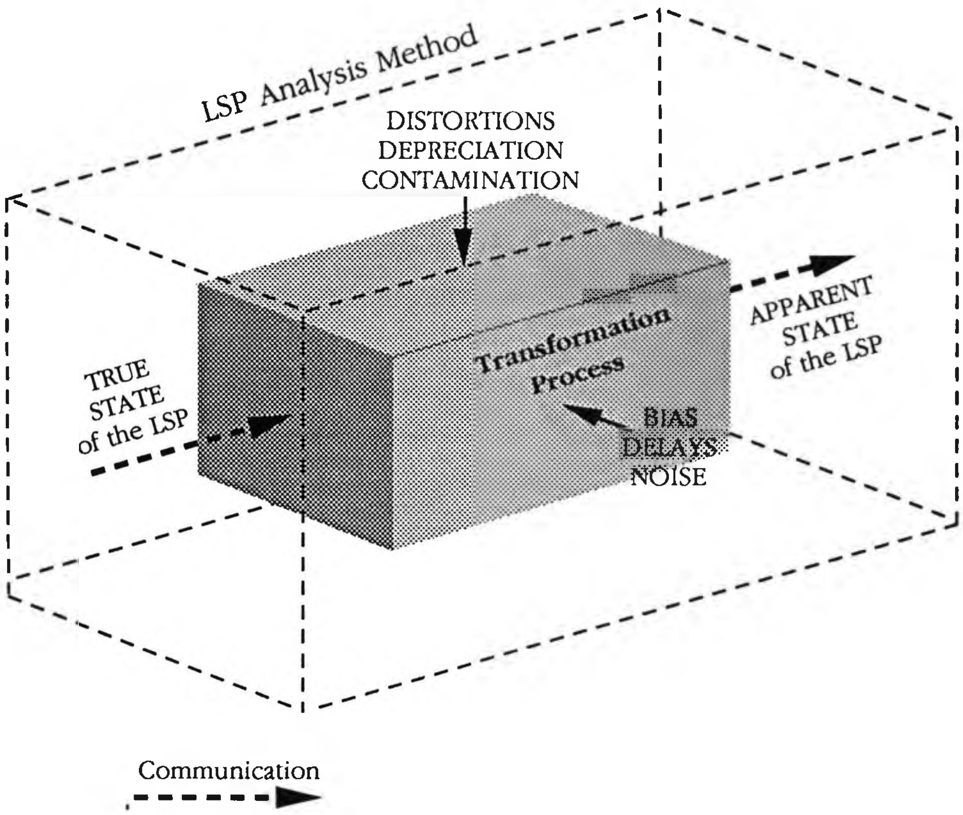
SoI itself are endogenous. Any simulation must reflect both these noises and disturbances. A LSP (the SoI) represented as a network schedule, can be shown as a simple black box, see Figure 4.1.



Simulation model as a MSoI of a LSP represented by a network schedule
Figure 4.1

A SoI is made up of components, variables, parameters, functional relationships, constraints and criterion functions (Shannon, 1975). Environmental exogenous influences effecting the SoI and influenced by the Meta–System of Interest (MSoI) should be included as a Wider System of Interest (WSoI), (Flood and Carson, 1988). These, together, will be the holistic model subsequently created.

There is likely to be a difference between what the Real World (RW) actually is and what it is perceived to be (i.e., isomorphic representation is unlikely to be foreseeable or desirable), with the additional blurring of what can be, and what is, modelled. Figure 4.2 below shows why there is a difference between ***true*** and ***apparent*** states (in this example a LSP) caused by the transformation process between the two. This change of state is caused by interference, some controllable by the analyst (for example, the method of analysis) and others due to the environment of the RW situation. Acknowledgement of this situation and how it can be taken into account is discussed further in Chapter 8.



Environmental disturbances between true and apparent states of a LSP caused by the analysis method
Figure 4.2

4.3 WHAT ARE MODELS?

Models are abstractions of what reality *might* be i.e., they are not isomorphic representations, they are able to make suggestions about what reality is like without being part of that reality. Information transformed into a model must represent the RW in a way which relates to the purpose for which it will be used. Figure 4.3 shows the relationship between a RW situation and its model, (Thesen, 1974: 146). Differences between the RWS and a model have to be acknowledged by the modeller so that he is able to determine what significant elements have to be included. He is unlikely to be able to produce an isomorphic representation without producing a replica—a pointless and impossible exercise in a dynamic and industrial situation.

A general model definition can be considered as:

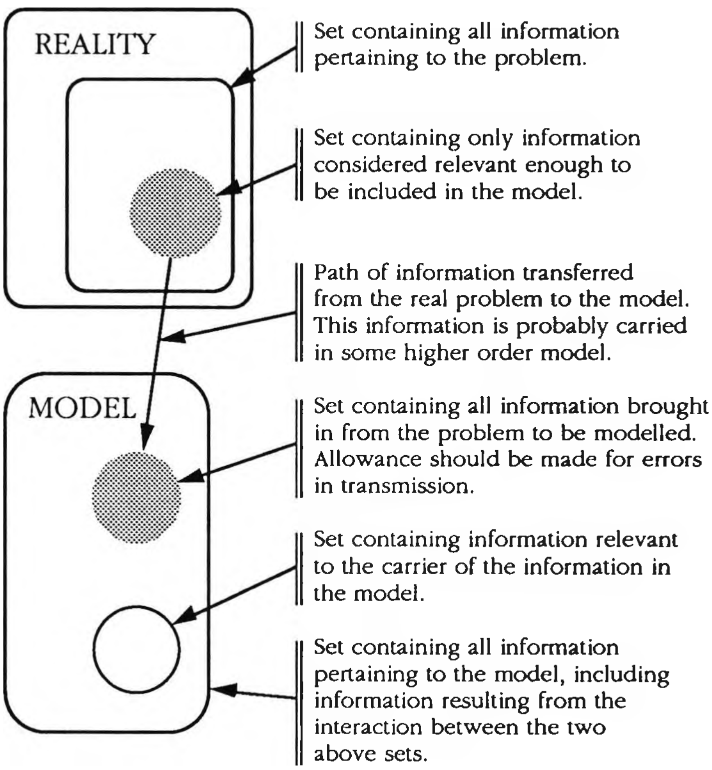
> ...a qualitative or quantitative representation of a process or endeavour that shows the effects of those factors which are significant for the purpose being considered.
>
> (Chestnut, 1965: 108)

Thesen (1974: 145) defines a model as:

> ...a real or imaginary entity containing information in a certain predefined form, intended to be interpreted by its user in accordance with certain predefined rules.

For the purposes of this systemic research, the following is used:

> A model helps to provide an understanding of a real world situation, utilising an abstract medium, with information obtained by a set of iterative rules.



Relationship between the real world and its model
Figure 4.3

Such a set of iterative rules could be in the form of a methodology, examples being:

(1) Formulating the problem;

(2) Constructing the model;

(3) Testing the model;

(4) Deriving a solution from the model;

(6) Implementing the solution.

Ackoff (1962: 26)

or:

(1) Establish the problem;

(2) Observe first qualitatively, then quantitatively;

(3) Formulate hypothesis about the unknown real situation;

(4) Experiment to test the hypothesis;

(5) Decide on the hypothesis (accept–reject);

(6) Finally [formalize] the outcome in managerial terms.

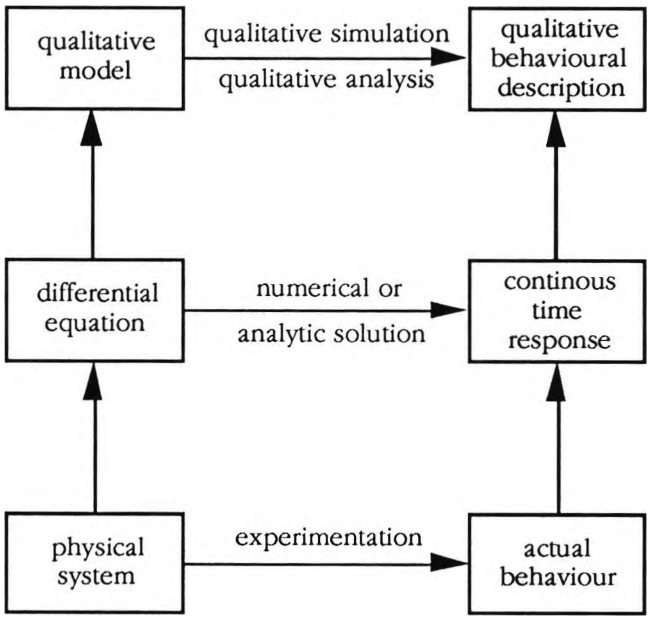
Chorfas (1962: 176)

The specific methodology, that of System Dynamics, is discussed in Chapter 5.

4.4 Real World Situation as a Model

If a large RW situation is represented by a model, there is likely to be a problem concerning detail to be portrayed by that model:

> Every model is necessarily an abstraction of the real world; it is because it is an abstraction, that the model is useful. The key question in system modelling is to determine the most appropriate *level of abstraction* (model) for a given task. (Leitch, 1987: 246)



Three tier abstract hierarchy of models
Figure 4.4

Levels of resolution involved can be divided into a three tier hierarchy, see Figure 4.4 (Leitch, 1987: 247).

A possible solution to RW analysis is by "chunking" (Klir and Valach, 1967; Simon, 1969a) i.e., utilising the "black box theory". By giving consideration to input(s) and output(s) only, the transformation process itself can be ignored. This simplifies a system made up of sub– and sub–sub–systems, but reduces determination of system outputs (Holstadler, 1979). For example, if a "black box" transforms two inputs into eight outputs, one is unaware of the transformation being either a multiplier of four, or two times two. This could make future changes to the transformation process difficult, because if a future multiplier (within the information process) is increased by one, the output could be either ten or twelve.

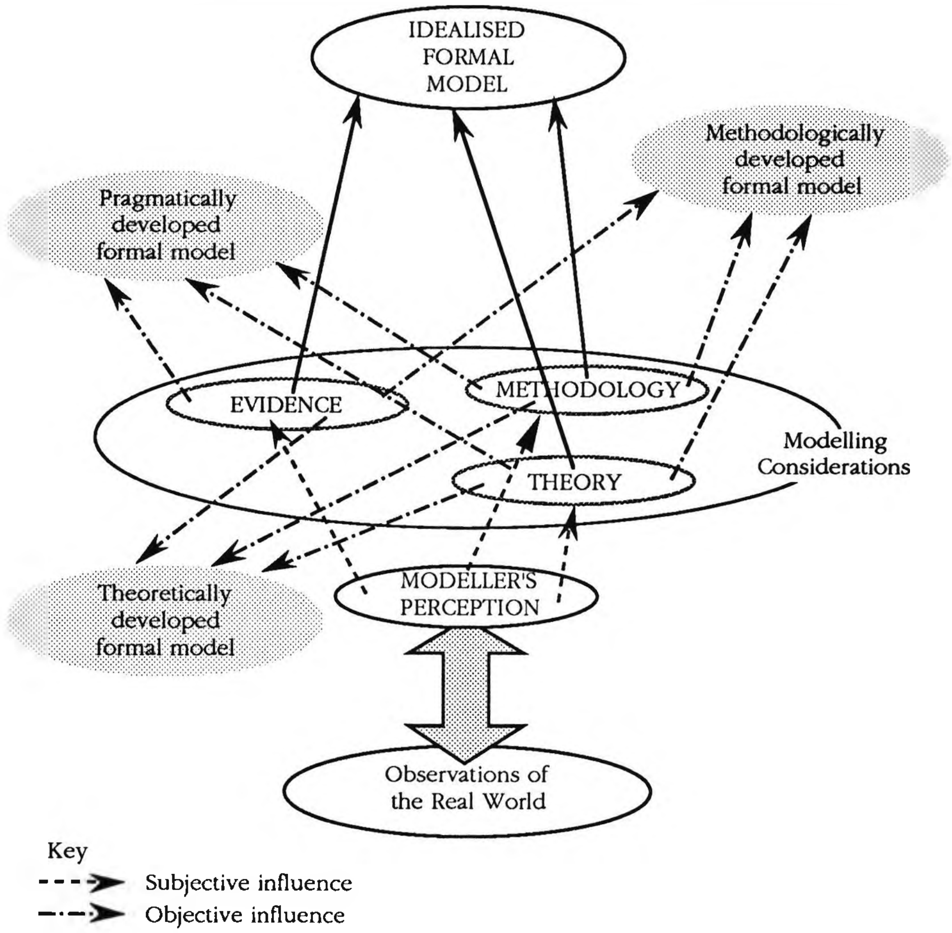
4.5 Formulation of a Formal Model

4.5.1 Introduction

Formulation of a formal model relies upon the interdependence and influence of methodology, evidence and theory and by interpretation or the modeller's perception, see Figure 4.5.

A modeller's perception of his observation of the RW (cognition) will have subjective influence on the evidence he finds (or is given i.e., the questions he asks in order to obtain such evidence), the theory he develops and methodology he selects. Resultant models will be an objective function of these three components. The analysis stage in development of a model centres on the modeller examining a RWS. Conceptualisation considers verbal models given to him as evidence from which a mental model is derived. By utilising a methodology, verbal and mental models are synthesised into a Formal Model. This would prevent: 'Model building...become[ing] an end in itself, and many models have only a distant relationship to ill–defined questions.' (Rothkopf, 1973: 60)

When considering Figure 4.5, achievement of an Idealised Formal Model (IFM) is possible if there is a balance between evidence available, methodology used and theory utilised. Should any one of these be unfairly weighted (whether consciously or not) then ensuing models will be biased.



Formulation of an idealised formal model
Figure 4.5

4.5.2 PRAGMATICALLY DEVELOPED MODEL

Pragmatically Developed Formal Models are the result of a modeller being unable to collect relevant data or interpret what he finds/sees. Pragmatical analysis is one concerned with matters of *fact* rather than *theory*. The modeller produces a model based on evidence he interprets. As a brain's internal model/perception can be confused by what is seen, and without a means of interpreting evidence by use of theory and methodology, a flawed model can result. An example illustrating this is two objects of dissimilar heights and at different distances away, both appearing to be the same height. When a modeller models, by drawing for example (two dimensional), the objects will have the same height if it cannot be related to any other object. Had a methodology been used he may have walked towards the nearest object (as part of the analysis stage) and then realised that it was becoming disproportionately taller (by taking into account a third dimension). Likewise, if the analyst had been aware of the theories of perspective, he may not have believed that because two objects appear to be the same height, they are actually so.

4.5.3 THEORETICALLY DEVELOPED MODEL

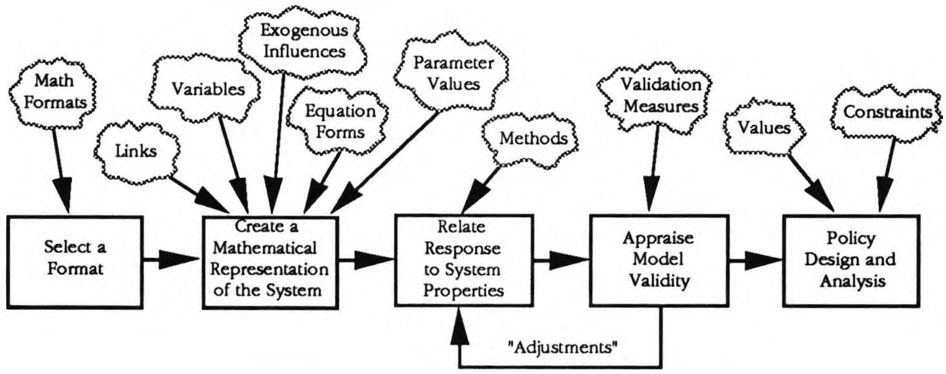
Theoretically Developed Formal Models can result from a misunderstanding in interpreting certain RW situations. Misinterpretation could be due to the "purest" nature of theoretical work. The value of such work, in this instance, is that it forms a "pure" base or datum from which to work. In such a sphere, models and/or ideas can be developed at epistemological or ontological levels of knowledge—some may be subjective, some unchallengeable and many impractical. It is accepted that these are deep philosophical issues and it is not necessary to discuss them here. However, it is acknowledged that for practice to start and continue successfully, a theoretical foundation needs to be determined.

Theoretical bias of investigation, without the balance of including a methodology, both related to potential evidence, is likely to produce a model impractical and unsuitable to use outside a theoretical investigation. An example of a theoretical model could be one developed to determine the means of motivation within an industrial environment. Such a model could show how money is the prime motivator. This stance would therefore ignore the workforce's other needs (of which financial reward is just one facet).

4.5.4 METHODOLOGICALLY DEVELOPED MODEL

Bias caused by a Methodologically Developed Formal Model is perhaps a more contentious issue, for example the merits and use of Soft rather than Hard System's Methodologies and vice versa. A methodology may cause the modeller to make idiosyncratic or restrictive assumptions about the RW he is trying to model. In this situation the bias is caused, for example, by only considering quantitative data when utilising a Hard methodology, giving less credence to qualitative aspects of a RWS. The issue of methodologies is not the subject of this research, other than consideration being given to the effect of employing an appropriate one correctly and in a neutral manner, (Stevens, 1990; 1991).

Use of a methodology is a means of structuring a RWS to enable greater understanding to be achieved. Figure 4.6 (Starr, 1980: 51) identifies a model structuring and analysis phase with decisions that may be involved. At each stage of a general methodology, there is a need to incorporate various activities and/or considerations. Activities carried out during the use of a methodology are represented by blocks and methods or entities that could be incorporated into each activity, represented by clouds.



Model structuring and analysis decisions
Figure 4.6

4.6 WHY MODEL?

4.6.1 INTRODUCTION

Below are three significant points/reasons why it is desirable to model situations (in this case a LSP)—each may have a different end use, which in turn could affect the model building process:

- If a model considers all variables effecting the SoI (the MSoI) then this holistic approach leads to greater understanding of a project;
- Enables a Project Manager to test out feedforward scenarios before implementation of decision variations;
- It is less costly to run a computer model than use the project being considered itself as a test bed for decision variations.

There are various model classifications within the Formal Model classification of Schematical, Physical, Role–Playing and sub–divisions of Symbolic. These are identified by Fishman (1973) as those that can influence how an SoI will be modelled. Below is a brief explanation of model influence classifications.

4.6.2 ANALYTICAL AND NUMERICAL MODELS

ANALYTIC

Solution to a problem is provided by using its mathematical representation, for example Ohm's Law;

NUMERICAL

Complete solution is not possible, but by using numerical parameters on the model, a partial answer is possible, for example numerical integration.

4.6.3 DETERMINISTIC AND STOCHASTIC MODELS

DETERMINISTIC

Entities have mathematical or logical relationships to each other, therefore they determine the outcome, for example a true critical path method;

STOCHASTIC

Entities (at least some) have random variations, therefore relationships have to be determined on an average basis (for example Monte Carlo or PERT network schedules).

4.6.4 SIMULATION MODELS

SYSTEM SIMULATION

Examines simultaneous interactions of all parts of a situation. Enables a more complete analysis of a complex project that is not possible analytically;

IDENTITY SIMULATION

Model representation by use of symbols, but is limited due to the exclusion of many external disturbances that will/may affect outputs from a simulation. This simulation invariably takes as long as the real world situation;

QUASI–IDENTITY SIMULATION

Tries to emulate the real world situation, but excludes those elements that make identity simulation possible;

LABORATORY SIMULATION

Includes essential elements of the real world situation. This is sub–divided into two classifications: Operational Gaming and Man–Machine:

- **OPERATIONAL GAMING**

  Computers use information to aid in analysis of a situation and then becomes an opponent in a game to test human players;

- **MAN–MACHINE**

  Humans have an input into how much information is used by a computer to test the balance between human and machines to operate a situation;

COMPUTER SIMULATION

Totally computer controlled where all behaviour is programmable and follows a logical and predetermined sequence. However, any simulation experiment needs to have an element of external control so that sources of variation can be altered (interactive computer simulation). Computer simulations use a 'next event' approach to time advance, i.e., the next state change (Fishman, 1973; 1978). Such simulations can be either discrete or continuous. This later type generally excludes inactive time found in a real world situation.

The computer simulation model used inside PROMISS is founded on System Dynamics. This type of model has a mathematical foundation based on calculus, namely numerical differential and integration equations. These are time based dynamic systems derived from the following formula, (Zeitz, 1987: 466, translated from the original text written in German):

as a differential equation:

$$\frac{dx}{dt} = f(x, z, u), \quad \text{when } x(0) = x_0 \qquad \textbf{(4.2)}$$

as algebraic equations:

$$0 = g(x, z, u) \qquad \textbf{(4.3)}$$

$$y = h(x, z, u) \qquad \textbf{(4.4)}$$

From these equations the dynamic process of a conditional vector: $x(t) \in R_n$ and $z(t) \in R_m$, with respect to system input $u(t) \in R_p$ and a starting value of x_0, output can be determined. This resulting vector will be $y(t) \in R_q$. All information concerning mathematical models of the RWS can be determined by vectors *f*, *g* and *h* with dimensions *n*, *m* and *q*.

Dynamic system simulation utilising integration of ordinary differential equations (4.2), will have different computer algorithms to those for solution of algebraic equations (i.e., 4.3 and 4.4). Solutions to differential equations however, are better suited to *analog* (continuous) rather than *digital* (discrete) computers. Results from the latter remain acceptable for the purposes of this research, see Section 5.5.3—Euler's Method, for solving differential (difference) equations.

Solution of numerical integration equations with digital computers is undertaken by converting into algebraic approximations derived from transposing differential equations (4.2) to equivalent integral equations:

$$x(t_{k+1}) = x(t_k) + \int_{t_k}^{t_{k+1}} f[x(\tau), z(\tau), u(\tau)]\, d\tau \qquad \textbf{(4.5)}$$

Taking into account time steps:

$$t_k = k.h, \quad k = 0, 1, ... \qquad \textbf{(4.6)}$$

with system size of approximate values:

$$t_k \approx x(t_k), \quad z_k \approx z(t_k), \quad u_k \approx u(t_k)$$

As dynamic systems involve a change in time steps with variable $h = Dt$ (or DT, Δ—see Section 5.5.3), this can be adjusted to improve accuracy. Although Euler's (single step) and Runge–Kutta (multiple step) solution methods are discussed in greater detail in Chapter 5, the following should be noted:

- Euler's Method uses only functional values in a time interval $h = t_{(k+1)} - t_k$; for the approximate of the integral $f[...]$.
- Runge–Kutta Method uses small steps based on earlier functional values at intervals $t_{(k-1)}, t_{(k-2)}, ...$.

These methods extrapolate a solution $x_{(k+1)}$ for time steps $h \rightarrow 0$ from an asymptotic (a line that continually approaches a curve but never meets it) derived as a result of length h.

Main characteristics of numerical integration computer simulation programs are (Zeitz, 1987: 467):

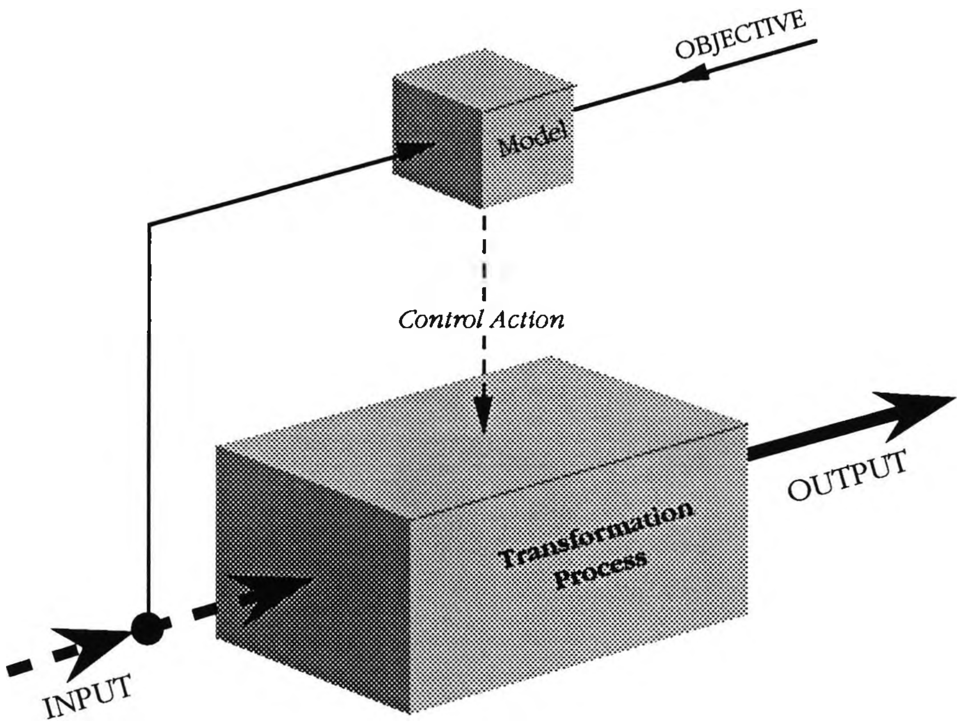
- Order of process or global misrepresentation r:

$$\sum_{i=1}^{k} |x_i - x(t_i)| \approx h^r \qquad \textbf{(4.7)}$$

- Explicit or implicit form of integration formula;
- numerical stabilisation process dependent on h and dynamics of the system;
- fixed or variable step length h;
- fixed or variable order of procedure r;
- necessary computation described by number of functional conclusions, dependent upon time step periods determined by $f(x, z, u)$.

4.6.5 MODELLING FOR CONTROL

By only considering behaviour that is programmable, those situations portraying behaviour that cannot be (or are at least difficult to) quantify are excluded. Use of a model to predict RWS behaviour is based on analytic control theory (Leitch, 1987). Making use of a simulation model introduces a means of *feedforward* control and is illustrated in Figure 4.7 below.



Use of a feedforward model
Figure 4.7

Feedforward control, for use in LSPs, can be defined as follows: In response to a measurable disturbance *(e)* on the LSP *(P)*, a manager/controller *(C)* responds in such a way as to eliminate the effect of the disturbance *(E)* on *P*. Thus, the effect of the error on the controlled output can be represented by:

$$y = (E + P.C)e \qquad \textbf{(4.8)}$$

where y is the effect of the error on the controlled output. Predicted error can be removed if:

$$C = -E.G^{-1} \qquad \textbf{(4.9)}$$

E can be assessed by use of a simulation model, (Flood and Stevens, 1987).

By considering Ashby's Law of Requisite Variety (LRV) a model used for control purposes has to be either explicitly or implicitly based on the controlled process, (Conant and Ashby, 1970; De Raadt, 1987). Shortfalls in control variety can be caused by insufficient information of RW elements being included in a model, and/or the conceptualised model structure being unsound. Leitch (1987) suggested that deficient model information can be compensated for by introduction of a 'tight' feedback system. With physical systems this is more easily achievable. When considering Forrester's (1961) multi–causal feedback systems, slower feedback paths found in a non–physical RWS are more likely to be subjected to environmental noise. The notion of feedback/feedforward control is discussed further in Chapter 7.

4.6.6 LSP Basic Characteristics

LSPs have two distinct basic characteristics (after Palaniswami, 1987):

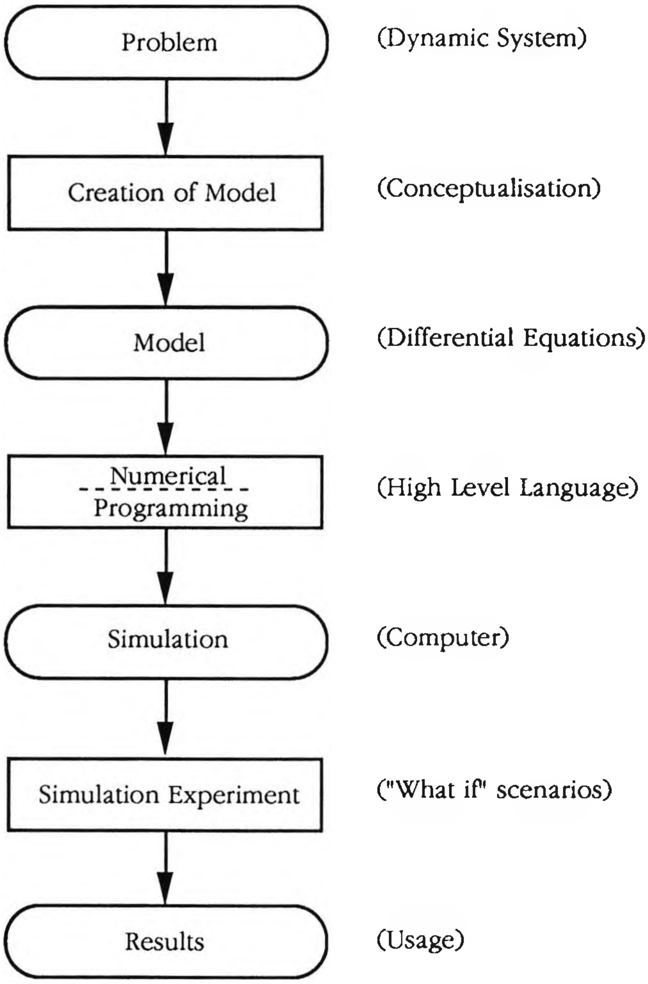
- Structure of situation—how logic and composition of activities making up a LSP are determined;
- situation behaviour—how a LSP behaves over time.

Both these two points *must* be taken into account and referred to when developing a model, as a representation of a LSP. Chapter 5 considers some of these issues when discussing the System Dynamics Methodology (SDM).

4.7 A Computer Simulation Methodology

This section considers a conventional methodology (rather than the SDM used in development of PROMISS's mathematical simulation model) as a means of developing a computer model. Zeitz (1987) surveyed various methods of developing computer simulation models. To ensure that an analyst is guided through a modelling process some form of "map" or methodology should be used. This takes an analyst from a dynamic problem (as perceived) through to a simulation model from which "what if" scenarios can be tried, see Figure 4.8.

Once a problem situation has been identified, initial model creation i.e., conceptualisation, can take place. This stage could include digraphs or causal loop diagrams (see Chapter 6), as used in SD. Conventional computer simulation will be founded on numerical equations derived from those



Steps through a computer simulation modelling methodology
Figure 4.8

described in Section 4.6.4. In a simulation program the following need to be considered (Zeitz, 1987: 467):

Simulation Model

Calculation of $f(x, z, u)$, $g(x, z, u)$ and $h(x, z, u)$ (from Equations 4.2, 4.3 and 4.4).

Numerical Model

Integration of differential Equation 4.2 and algebraic Equation 4.3 or the simultaneous solutions of both.

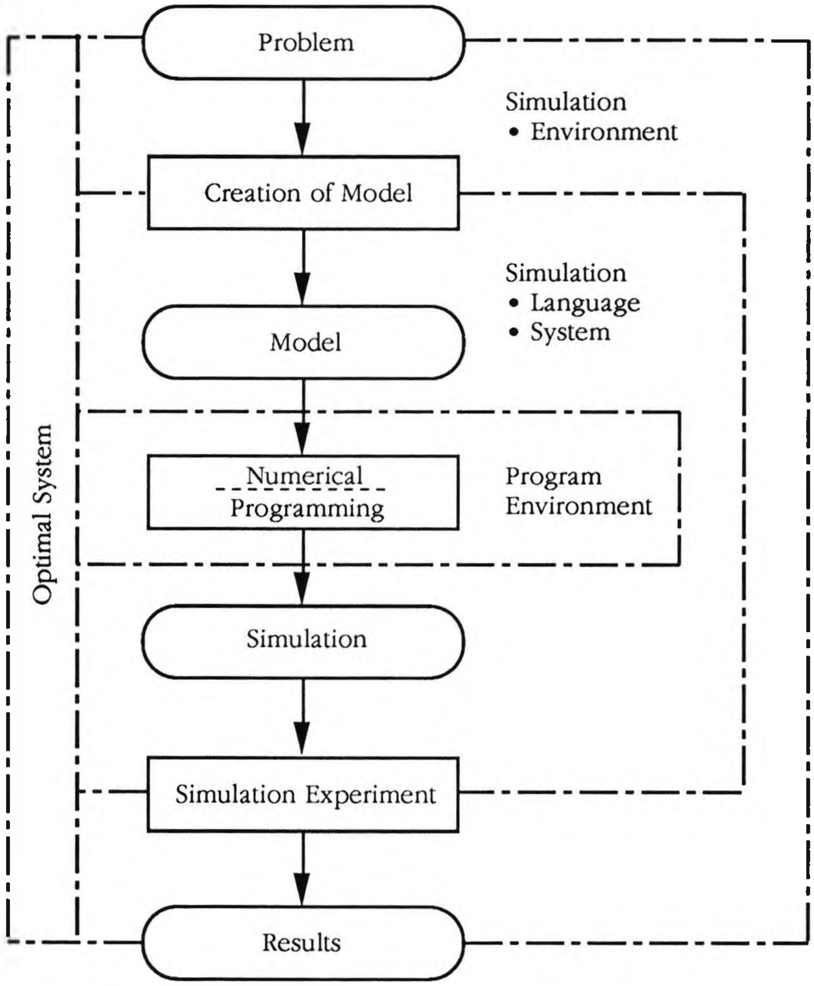
Displaying Results

Presentation and analysis of simulation results and possible transfer to other programs or external process (for example, control purposes).

Simulation Control

Intialisation; start/stop; changing, interacting; testing, etc. by a user of simulation programs and possible communication with other programs or external process.

Techniques related to programming will be dependent upon the initial problem being analysed and software being used. Many simulation models use a proprietary "shell" or software containing a number of routines enabling users to input, in simple terms, variables making up a situation to be modelled, see Appendix 1. PROMISS was programmed in Turbo Pascal, a high level computer language, but developed from models written in Micro–DYNAMO and Stella, see Chapter 5. Figure 4.9 shows how stages of a methodology are aided by various computer techniques (Zeitz, 1987: 467).



Simulation tools used in a simulation modelling methodology
Figure 4.9

Programming with proprietary software has advantages that, like Fourth Generation Languages (4GLs), users do not have to have any programming skills (although they do have to have a clear idea what they want and can achieve!). A distinct advantage with this means of programming is ease of producing graphical outputs—an asset in a DSS and/or showing results to people outside of those involved in modelling (it can also help those inside the development team). Users of this type of software do not have to write mathematical/numerical subroutines as many libraries and modelling aids can be purchased "off-the-shelf". Programming environments and simulation systems are not common or transferable across a wide spectrum of computers. Such difficulties mean that many simulation models already proven in one situation cannot be used in others. This prevents an analyst (and computer programmers) transferring skills from one problem being modelled to another, because computer environments and languages differ from client to client. However, use of Personal Computers (PCs) with common operating systems (DR/MS/PC.DOS), enable programmers to build up (and purchase) extensive libraries for use in more than one situation. In addition use of a high level language such as 'C', which is portable can be appropriate for program development.

If a simulation model can provide transferability and clear outputs, then both clients and analyst/modellers will be able to understand the dynamically modelled problem situation, through the use of a computer simulation. As with any "problem understanding methodology" users have to be aware of the limitations of the technique. For example with computer simulation, there is a reliance on an intermediary (computer programmer) with skills probably not understood by a client. Chapter 5 discusses SDM and the criticisms of such a methodology are similar to the criticisms of computer simulation in general.

4.8 VALIDATION

For any model to be of use it must be validated to determine whether or not it is representative of a RWS, and allowance must be made for the constraints imposed by the limited conditions it operates in. As any model is a simplified representation of the RW, validity is determined by a level of confidence. This can *only* be determined by historical behaviour, i.e., whether or not a modeller can say that his model is valid because it represents past perturbations of the RW. He cannot however *know* that the model is valid for the future, until the future becomes the present or history. Only after a period of these future-present-history cycles can confidence be increased in a model. Discourse on validation is continued in greater detail in Chapter 5.

4.9 CONCLUSION

This chapter has considered various aspects of simulation and modelling from the personal viewpoint of the author. It is acknowledged that these points are only one aspect of what is a large field of knowledge and viewpoints. Points raised in this chapter are applicable to the research and should be considered in this light.

A model can be distorted by both cognitive and/or development method/ structure (evidence–pragmatic; methodology–methodological; and, theory–theoretical) whether or not this is conscious, the end result will be "value distorted". The objective of developing a RWS model, through a modeller's perception, is clarification and explanation of that situation to others. In this research a mathematical based computer simulation is used for LRA scenarios. To develop an Idealised Formal Model requires a balance between three main factors which have to be considered during a systemic analysis namely, evidence, theory and methodology. Even if this balance is achieved, difficulties can be encountered in producing a representative model due to the dynamic environment of a RWS. The subject of this research is directed at a task (completion of LSPs) that can be positively identified due to changes to the work being contractually undertaken (i.e., Simple–Unitary, see Flood and Jackson, 1991). Changes in such work should be reflected in a PERT logic model for use by LRs. In turn this will change the integrated simulation model—thus going some way to deflect criticism of using such a modelling technique.

To develop such a dynamic computer simulation model this research utilises the specific methodology of System Dynamics. Although software normally used in the simulation stage is unsuitable, the foundation (both philosophically and mathematically) remains. The next chapter discusses System Dynamics and various issues associated with this methodology.

Chapter 5

System Dynamics

> Die Zeit, die ist ein sonderbar Ding, [Time is a strange thing]. An observation made by the Marschallin in *Der Rosenkavalier.*

> Industrial Dynamics is study of the information–feedback characteristics of industrial activity to show how organizational structure, amplification (in policies), and time delays (in decisions and interactions) interact to influence the success of the enterprise.
>
> Jay W. Forrester,
> *Industrial Dynamics,* 1961

5.1 INTRODUCTION

System Dynamics (SD) derives its roots from the major disciplines of mechanical, electrical and chemical engineering, (Doeblin, 1972). SD is now synonymous with Forrester's work at the System Dynamics Group, within the Sloan School of Management at the Massachussets Institute of Technology (MIT). The use of the SD Methodology (SDM) is a fundamental strategy used in this research. The objective is to lay a firm foundation and understanding of this Hard Systems Methodology (HSM), although there is a debate as to whether or not it should be classified as such, (Wolstenholme, 1982). Such debates do not detract from the aims and goals of this research.

From its conception (Forrester's definitive book was written in 1961) SD has attracted both positive and negative attention. Knowledge of SD outside the sphere of management and MIT, was initially due to Forrester's World models used by the Club of Rome in *Limits to Growth* (Meadows *et al.*, 1972; Cuypers and Rademaker, 1974). SD has still maintained some prominence within the sphere of creative problem solving, (Flood and Jackson, 1991; Stevens, 1990; 1991; Stevens and Flood, 1988; Wolstenhome, 1990).

This chapter firstly briefly overviews the SDM; secondly, considers two main themes of criticisms levelled against it as: a) a methodology and b) use of DYNAMO (DYNAmic MOdelling) computer simulation; and thirdly, identifies the usefulness of adopting such a methodology as an integral part of this research.

5.2 Dynamic Situations

As most RW situations are dynamic in nature, a methodology used in its analysis should reflect this characteristic, in other words, it: '...must examine the structure and dynamics...before the performance specification can be determined....' (Forrester, 1968c: 602). Computer simulation is one of the most effective means available for supplementing and correcting human intuition in such dynamic situations (Nordhause, 1973; Forrester *et al.*, 1974). Models derived from mathematics are a powerful conceptual device, increasing the role of reason in determination of a policy and analysis of future occurrences.

SD involves mathematically derived computer modelling as part of the methodology, considering complex continuous systems made up of dynamic components, (Forrester, 1961). From this, a set of computer coded equations are derived, representing dynamic (rather than static) behaviour in the form of flows and more importantly, their relationships. Response to various stimuli (both internal and external) can be modelled and monitored during the dynamic life of situations. The life span of such models is only limited by knowledge available at the time of model conceptualisation.

5.3 System Dynamics Methodology

5.3.1 Introduction

For a RW situation to be analysed and eventually simulated on a computer, to enable a greater understanding to evolve, some form of methodology is used:

> The creation of a complex simulation is based on the proper integration of several interdependent factors. While simulationists do not agree on the exact terminology to be used in presenting these factors, they do tend to agree that there are some concepts which are crucial to the modeling process. Research also indicates that simulationists use some sort of modeling process in order to organise their activity. (Banks *et al.*, 1987: 13)

SDM is specifically an endogenous one unlike many others which take an exogenous viewpoint. A boundary is drawn around the System of Interest (SoI), so that all the components necessary to generate problem behaviour are contained within it, *including* the environment, represented as a Meta System of Interest (MSoI):

> The cause and effect relationships between environment and system are uni–directional, whereas the internal elements are structured into feedback loops that cause the internal elements to interact. The environment can effect the

> system, but the system does not significantly affect the environment. (Forrester, 1969)

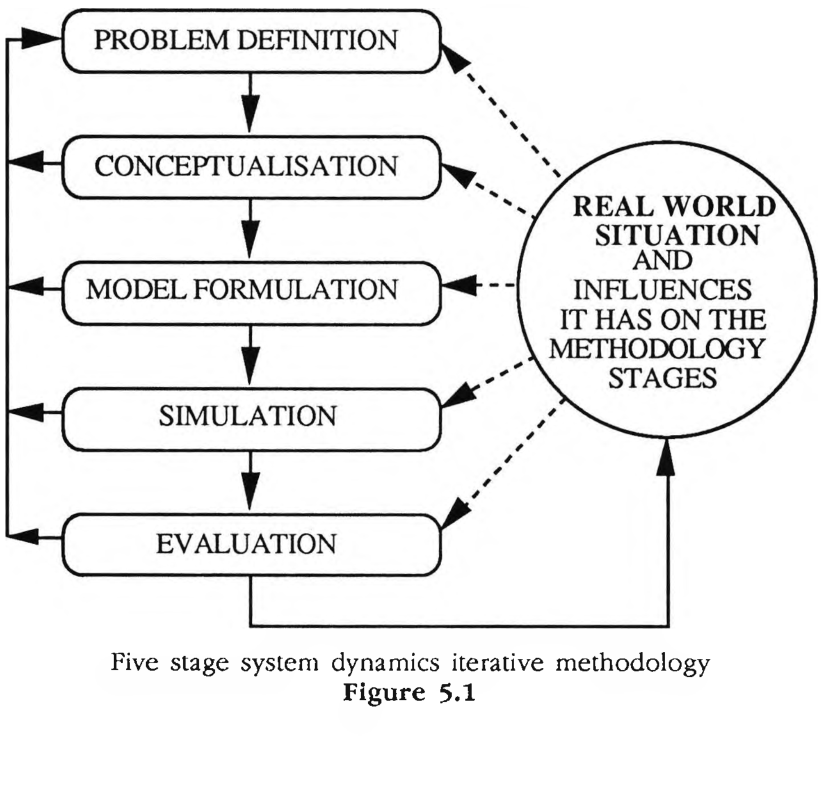
SDM, as a HSM, is objective in nature and makes use of mathematical computer simulation as a modelling medium. It is able to provide a method for explaining a situation, with subsequent models being utilised for policy testing:

> ...the system dynamics process of problem/symptom explanation, qualitative analysis and recommendations for change,...constitutes a general methodology for system enquiry. This is essentially a further adaptation of pure science research methodology but with the following advantages:
>
> 1. general applicability and independence of content enquiry;
> 2. an ability to provide guide–lines at *every* stage;
> 3. sufficiently flexible to function at the higher levels of the system spectrum.
>
> (Wolstenholme, 1982: 554)

Bearing these points in mind, the following section briefly describes the SD methodological process. A detailed discussion continues during the rest of the chapter, giving additional clarification.

5.3.2 THE METHODOLOGY

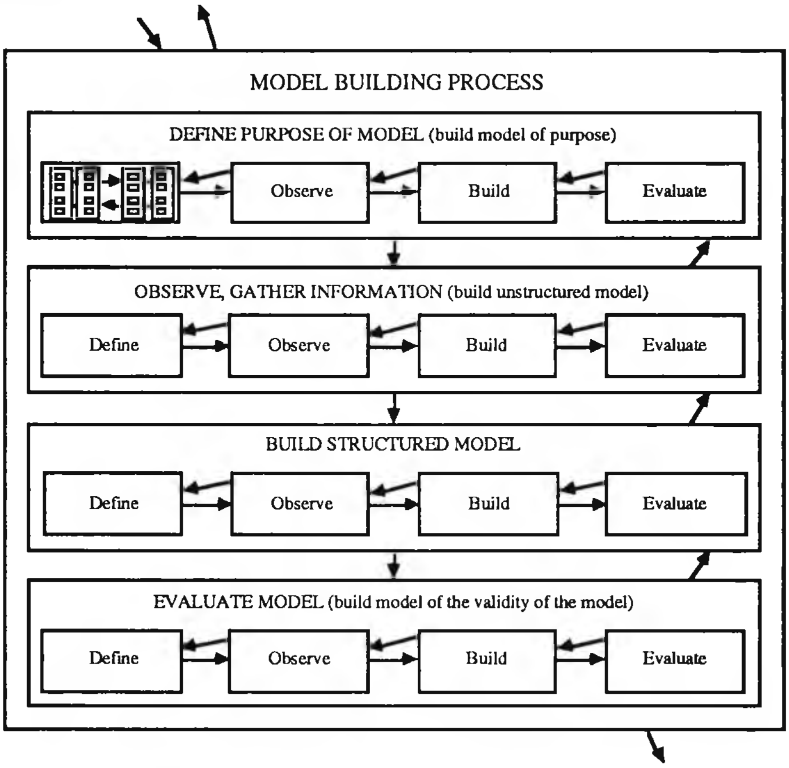
The SDM is generally accepted to be a five stage iterative process, following a similar format to those given in Section 4.2, see Figure 5.1:



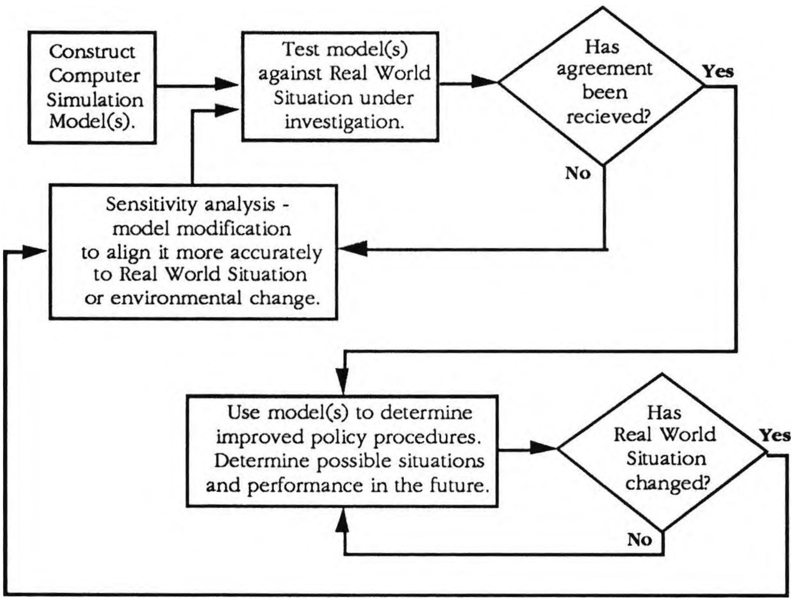
Five stage system dynamics iterative methodology
Figure 5.1

For a such a methodology to be successful, each stage needs to be iterative. Thesen (1974: 150) goes as far as introducing a recursive procedure for each stage, see Figure 5.2.

The methodology needs to be used in conjunction with a decision algorithm in an attempt to ensure the subsequent model(s) is a "good fit", such as that shown in Figure 5.3.



Systems approach to model building process
Figure 5.2



Modelling decision algorithm
Figure 5.3

This procedure, within the methodology, attempts to steer an analyst and model builder through the difficulties of having to compromise between sufficient information and model complexity:

> Until recently the systems which were studied were sufficiently simple that after all of the irrelevant variables were discarded, the number remaining was small enough to give a manageable model. When genuinely complex systems are tackled, however, the old procedure does not work; either one is forced to discard relevant variables (or interactions) to get a model of manageable complexity which then is of poor quality, or else one ends up with a model which is of good quality but itself is unmanageably complex. (Thesen, 1974: 150, credited to R. C. Conant, *Ph.D., Dissertation*, University of Illinois, Urbana, Illinois, 1967: 1)

5.3.3 FIRST STAGE: PROBLEM DEFINITION

Setting of a SoI boundary, together with level of analysis resolution, theoretically sets the number of elements, feedback loops and influences to be analysed. In practice this is difficult. Legasto and Maciariello (1980) refer to this as a dichotomy: a subjective balance between problem definition and scope (with the level of resolution).

The problem definition stage of the SDM is considered as being most important, because of requirements for setting a level of analysis resolution, (Leitch, 1987), i.e., detail needed to be included within the SoI boundary. Forrester (1968b) considers that positioning of the boundary is dependent upon specific SoI behaviour being studied. All elements likely to generate an influence on a situation *must* be included. It is this criterion that *has* to be used when deciding what will be included and excluded from a model. Such an internal view creates a dramatically different problem focus—the search for feedback structures that can create or exacerbate the problem of system behaviour, (Coyle, 1977).

5.3.4 SECOND STAGE: CONCEPTUALISATION

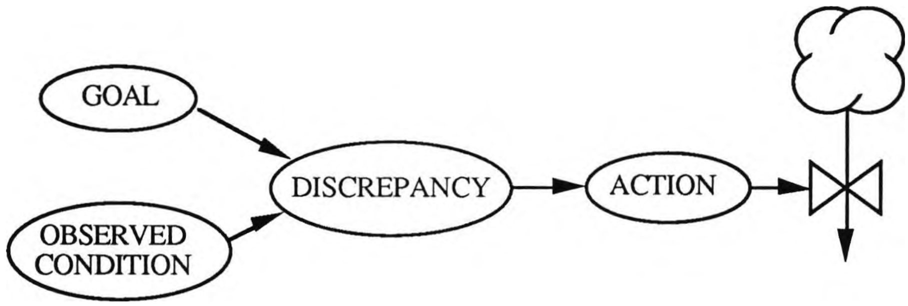
This stage involves identification and description of a SoI within a defined boundary by making use of causal loop diagrams or digraphs. This forms a foundation for identifying system structure and feedback loops. These diagrams show how elements (variables) influence and interact with others.

The SD approach searches for causes and cures of a problem within a defined boundary, making the focus internal, not external. These situations contain, or are brought about by, the notion of feedback, i.e., '...a closed sequence of causes and effects, a closed path of action and information.' (Richardson and Pugh, 1981: 4).

> The feedback loop is fundamentally a closed process in which a decision through time delay and distortion, influences the state of the system which, after further delay and distortion is detected as the observed state of the system. The focus of attention is on how this loop operates.... The boundary [drawn around the SoI] implies dynamic independence in the sense that any variable crossing the boundary from the outside is not itself a function of the activity within the boundary. (Forrester, 1968b: 407)

5.3.5 THIRD STAGE: MODEL FORMULATION

After a system structure has been clearly defined and stated, the SDM differs from other methodologies in that links between elements are considered as flows. These flows connect the two main SD element classifications, Rates and Levels.



Components of a rate equation
Figure 5.4

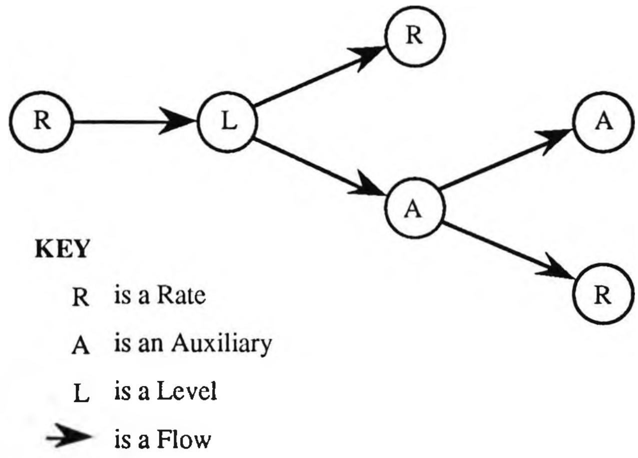
RATES

Rates are system flows and depend upon preceding levels and constants only. These equations are policy statements and in general terms, contain four components: an observed condition; a goal; and, a discrepancy, leading to action, see Figure 5.4, (Forrester, 1968a: 4–15).

LEVELS

Levels are system state indicators and accumulate flows directly from Rates. They also decouple system Rates, permitting flows to be independent of each other, but still able to have their effect measured on a Level. Rates feeding into these Level equations are a means for advancing the simulation into the future.

Auxiliary equations used in SD modelling are sub–divisions of Rate equations. Three main advantages for this are: 1) they clarify and break down Rate equations; 2) they are able to change or unify the measurement medium; and 3) are able to be used as a Table function to give a more precise control of subsequent Rate equations, (Uys, 1984). The logic of SD equations are shown in Figure 5.5.



Logic of SD equations
Figure 5.5

5.3.6 FOURTH STAGE: SIMULATION

This stage is discused in Section 5.5.

5.3.7 FIFTH STAGE: EVALUATION

This stage is discused in Section 5.10.

5.4 SYSTEM DYNAMICS AS A THEORY OF STRUCTURE

5.4.1 SYSTEM STRUCTURE

SD philosophy considers behaviour (or time history) of a situation as being principally caused by the SoI's structure, 'Industrial dynamics [System Dynamics] is a theory of structure of systems and of dynamic behaviour in systems, not a theory limited to the description of a single system....' (Forrester, 1968c: 604). Structure includes not only the physical aspect of plant and production processes, but more importantly the policies and traditions, both tangible and intangible that dominate decision making. Structure is seen as having four significant hierarchies, Forrester (1968b: 406).

These systems have four main characteristics: '..order, direction of feedback, nonlinearity, and loop multiplicity' and form "four significant hierarchies of system structure," (Forrester, 1968b: 402):

The Closed Boundary
The Feedback Loop as the Basic System Component
Levels (the integration, or accumulations, or states of a system)
Rates (the policy statements, or activity variables, or flows)
Goal
Observed Conditions
Discrepancy between Goal and Observed Conditions
Desired Action

5.4.2 ORDER

Mathematically, with respect to calculus, the number of integrations (of first–order difference equations) determines the system order. In SD terms, this is determined by numbers of levels. As levels are also first–order equations, many managerial systems have at least twentieth and some cases in excess of one hundredth order.

5.4.3 DIRECTION OF FEEDBACK

This refers to loops being either negative or positive feedback; negative usually being an inhibiting or controlling influence and positive being an augmenting one creating either a growing or declining form of behaviour.

5.4.4 NON–LINEARITY

This is a system containing division or multiplication of variables and/or where there is a coefficient which is also a function of a variable. When there is non–linear coupling of positive and negative (alternatively all positive or all negative) there will be a dominance shift between loops (Forrester, 1968a). Steady–state control mathematics of linear systems are generally concerned with negative feedback, giving little acknowledgement of effect brought about by positive feedback.

Positive feedback influenced systems (like loops) produce an exponential growth or decline from a set point, but its presence is essential for system growth. In non–linear, unlike linear, systems such feedback is not necessarily detrimental due to control interaction of other affected loops within the system, (positive loops can control other positive loops to give a controlled system). Primary control is normally exerted by negative feedback.

5.4.5 LOOP MULTIPLICITY

Very few managerial, economic or social systems can be represented adequately by a single loop structure. Several loops, both positive and negative, are invariably involved. It is the number and interaction of these loops that make systems, their outcomes and identification of key variables, taking into account counterintuitive behaviour, difficult to comprehend without use of computer simulations, (Forrester, 1970).

5.4.6 SYSTEM FLOWS

SD considers systemic thinking as being viewed most effectively in terms of common underlying flows, instead of in terms of separate functions. This flow orientation causes the focus of attention to be made across inter–element and system boundaries—to take a holistic, rather than reductionist, approach. A meaningful system's framework results from tracing cause and effect chains through relevant flow paths.

5.4.7 SD SUBJECT SUMMARY

A subject summary is given in Figure 5.6 below (Wolstenholme, 1982: 548) below.

<table>
<tr><td rowspan="2">System Description
Qualitative Analysis</td><td colspan="3">Quantified Analysis Using
Continuous Simulation Techniques</td></tr>
<tr><td>Stage I</td><td>Stage II</td><td>Stage III</td></tr>
<tr><td>1. Of existing proposed systems.
2. In terms of system flows (using levels and rates) and objectives.
3. Using physical, cash and information flows.
4. To examine feedback loop structure.</td><td>To examine the behaviour of all system variables over time.
To examine the validity and sensitivity of the model to changes in:
(i) structure;
(ii) policies;
(iii) delays uncertainties.</td><td>To examine alterative structures and control policies based on:
(i) intuitive ideas;
(ii) control theory analogies;
(iii) control theory algorithms:
in terms of non–optimising robust policy design.</td><td>To optimise system parameters.</td></tr>
<tr><td>To provide:
(i) a perspective on the observed problem or symptom;
(ii) a qualitative analysis on which to base recom–mendations for change.</td><td colspan="3">To provide a quantified assessment of alternative ways of improving system performance.</td></tr>
</table>

System Dynamics: a subject summary
Figure 5.6

5.5 SYSTEM DYNAMIC MODELS

5.5.1 SYSTEM DYNAMICS AS A SIMULATION APPROACH

A common misunderstanding of SDM is that it is considered as '...being just a simulation approach, rather than a methodology based on ...a "systems theory" of the firm in much the same way Cybernetics is a theory of social behaviour.' (Ansoff and Slevin, 1968: 395). Utilisation of computer simulation models is a technique used *within* SDM and not the methodology itself:

> Indeed, many traditional system dynamics practitioners would find it incongruous to separate the methodology into the distinctive phases suggested...and argue that influence diagrams are drawn with the relationships and dimensions of quantified analysis in mind. However, whilst this statement is true, it is important to realise that the argument only tends to reinforce the perception of system dynamics as just a technique for simulation analysis. In general system enquiry terms it is just much more, and the simulation phase of system dynamics can alternatively be viewed as just in–depth expansion of the analysis phase of the methodology—that is, as a technique within the overall methodology. (Wolstenholme, 1982: 554)

SD modelling within SDM utilises a simulation technique considering non–linear, multi–causal loop systems, closely relating to situations found in RWS as being brought about by feedback. A policy or decision eventually has an influence on the state of that situation, which in turn affects future situations:

> The feedback loop is seen as the basic structural element of systems. It is the context within which every decision is made. Every decision is responsive to the existing condition of the system and influences that condition. (Forrester, 1968b: 408)

SDM makes use of DYNAMO, a mathematical computer simulation compiler initially used on mainframe computers, but latterly available in a micro format working in a Pascal (see Section 5.8), rather than the original Fortran environment, (Pugh III, 1983). There is also an English derivative DYSMAP (DYnamic System Modelling and Analysis Package), also running in a Fortran environment, developed by the System Dynamics Research Group at Bradford University, (Coyle, 1977; 1983; Wolstenholme and Coyle, 1980). SDM is not limited to use of these software packages, as any language able to represent system states and flows in the form of levels and rates will suffice (Roberts *et al.*, 1983b), see Appendix 2 for one written in BASIC (a simple subroutine) and Turbo Pascal.

Both these computer simulation languages are used with SDM and have been developed further into Professional DYNAMO and DYSMAP2 (further developed into DYSMOD—DYnamic System Model Optimiser and

Developer) to counter criticisms such as optimisation of variables (Burns and Malone, 1974; Keloharju and Wolstenholme, 1988; Wolstenholme, 1990). This later software package utilises a heuristic Search Decision Rule (SDR) as an optimisation algorithm.

5.5.2 MATHEMATICAL FOUNDATION OF SD COMPUTER SIMULATION MODELLING

The simulation stage of SDM utilises DYNAMO (or derivative) to simplify mathematical input and permit variables to relate to a more natural language for element/variable identification. Such a user interface enables the analyst to continue with situation understanding formulation, without being distracted by having to program in a high level computer language.

Mathematics of SD simulation are not of an analytical differential equation format, unlike pure mathematical models. DYNAMO represents a mathematical coding of SD flow diagrams, with mathematical conversions being undertaken by the software compiler.

Alternatives are pure mathematical forms of linear and nonlinear representations, (Starr, 1980):

Linear:

$$\dot{x} = Ax + Ba, \qquad \text{given}: x(0) = x_0 \tag{5.1}$$

Nonlinear, with respect to (t):

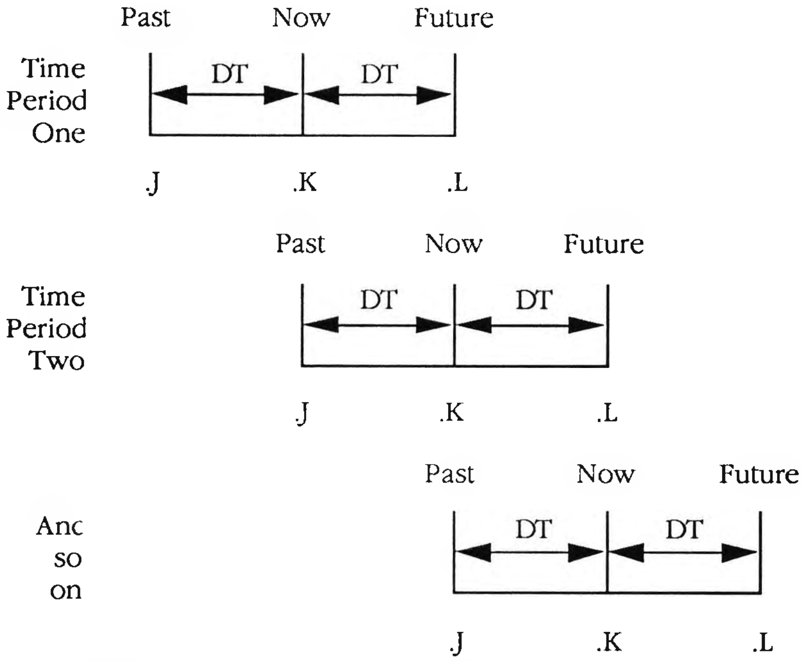
$$\dot{x}(t) = f[x(t), a(t)] \tag{5.2}$$

given: the initial value $x(t_0)$, for state vector $t = t_0$

where:

- x = n th order (dimensional) system state vector (i.e., Levels);
- a = exogenous input control vector (m–dimensional);
- A = parameters forming an $n \times n$ matrix;
- B = parameters forming an $n \times m$ matrix;
- $f()$ = n th order nonlinear vector function.

Models derived from SDM offer the advantage of providing an easy means of trying out policy changes and then *predicting* their outcome. A time sliced output in the future (time interval .L) could be predetermined now (time interval .K), by mathematically projecting from the past (time interval .J). This time step process is shown in Figure 5.7 below.



Time step process in SD modelling with DYNAMO
Figure 5.7

State indicators (Levels) are the critical and main variables of SD models and follow a format:

New Level = Old Level + Time Increment * (Inflow Rate – Outflow Rate) **(5.3)**

Utilising the time interval notation of .J, .K and .L, with DT as the time increment, Equation 5.3 can be rewritten:

$$L.K = L.J + DT * (IR.JK - OR.JK) \quad \textbf{(5.4)}$$

By considering Equation 5.4 with a single rate and τ as the time constant:

$$L.K = L.J + DT * \frac{R.JK}{\tau} \quad \textbf{(5.5)}$$

by moving L.J to the left hand side and dividing equation by DT so that as it approaches zero, it can be represented by an analytical solution of a continuous system:

$$(t) = \frac{l(t)}{\tau} \quad \textbf{(5.6)}$$

which can be rewritten in normal differential d*x*/d*t* notation, (Britting, 1976):

$$\dot{x}(t) = \frac{x(t)}{\tau} \tag{5.7}$$

The DYNAMO simulation modelling compiler is mathematically founded on difference or Step–by–Step equations, with most DYNAMO statements being of a first–order linear format (Equations 5.3 and 5.4). Their theoretical foundation (Burghes and Wood, 1980), is shown below:

$$x_{k+1} = Ax_k + B \tag{5.8}$$

when:

(k = 0, 1, 2, . . .) where A and B are constants and ($A \neq 0$) the solution to Equation 5.8 can be found:

$$x_1 = Ax_0 + B \tag{5.9}$$

$$x_2 = Ax_1 + B = A(Ax_0 + B) + B = A^2x_0 + B(1 + A) \tag{5.10}$$

$$x_3 = Ax_2 + B = A^3x_0 + B(1 + A + A^2) \tag{5.11}$$

so that:

$$x_k = A^kx_0 + B(1 + A + A^2 + \ldots + A^{k-1}) \tag{5.12}$$

$$= A^kx_0 + B\frac{(1 - A^k)}{(1 - A)}, \quad \text{when} : (A \neq 1) \tag{5.13}$$

when geometric progression on the right hand side is summated and letting A = 1, then:

$$x_k = x_0 + nB \tag{5.14}$$

The solution can be summarised by:

$$x_k = \begin{cases} A^kx_0 + B\frac{(1 - A^k)}{(1 - A)} & (A \neq 1) \quad (5.15) \\ x_0 + kB & (A = 1) \quad (5.16) \end{cases}$$

Solutions to the above two equations are dependent on x_0, A and B. If:

$$x_0 = \frac{B}{(1 - A)}, \text{ then } x_0 = x_1 = x_2 = \ldots = x_k = \ldots \tag{5.17}$$

and if:

$$A > 1, \text{ and } x_0 \neq \frac{B}{(1 - A)} \tag{5.18}$$

then solutions will grow unbounded i.e., large positive and negative values of x_k as time increases.

If:

$$0 < A < 1, \text{ then as } k \rightarrow \infty, \; x_k \rightarrow \frac{B}{(1-A)} \tag{5.19}$$

then conversely solutions will converge as time increases. If $-1 < A < 0$ then an oscillating behaviour will occur.

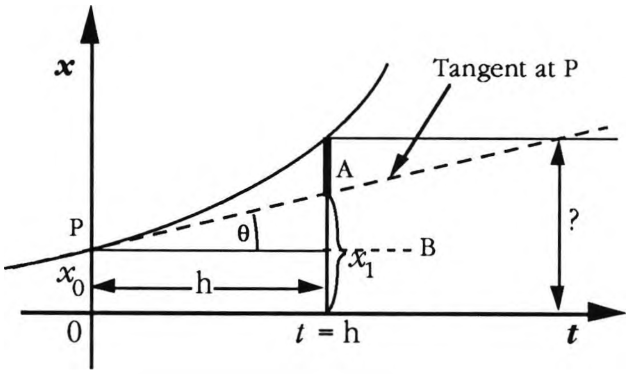
The specific mathematical format for dynamic modelling (DYNAMO and Stella) is either Euler's and/or Runge–Kutta Methods.

5.5.3 EULER'S METHOD

This Step–by–Step method is used to determine the trajectory of a curve plotted against axis x and t, given the differential equation:

$$\frac{dx}{dt} = f(x, t) \tag{5.20}$$

This can be shown diagrammatically in Figure 5.8.



Euler's Method for plotting trajectory of a curve
Figure 5.8

Given:

$x = x_0$ when $t = 0$, one needs to find x at some time later (?):

Taking:

x_1 as the approximation to x at $t = h$

$$x_1 = x_0 + AB \tag{5.21}$$

and:

$$\frac{AB}{h} = \text{Tan } \theta \tag{5.22}$$

which is the same as:

$$AB = h \text{ Tan } \theta \tag{5.23}$$

Considering tangent at P in Figure 5.8 and:

Taking Tan θ as the slope of the tangent and also $\frac{dx}{dt} \equiv \dot{x}$

therefore Tan $\theta = \dot{x}$ at P **(5.24)**

Euler's Method can be summarised and written as an equation:

$$x_1 = x_0 + h\dot{x}_0 \quad \textbf{(5.25)}$$

A decrease in this integration error is obtained by reducing the time step (see Section 5.5.5). Solutions using Euler's Method equations are numerical "approximations" i.e., there is a difference between these results and an "exact" solution—caused by an integration error.

Euler's integration method is considered as being suitable for continuous event computer simulation and works well in SD simulation models, (Pugh, III, 1980: 179). Characteristics of models requiring such mathematical solutions are:

1) Output accuracy of simulation is only of a moderate importance because in many situations in which SDM is used, generally only poor quality usable data and parameter estimates are available. Accuracy level is determined as the difference between solution of an integration calculation and exact answer (high being a difference of ≤0.1%);

2) there is a requirement for frequent plotted outputs. Accuracy and smoothness of graphical output is improved with a greater number of model outputs. A higher integration method, the Runge–Kutta Method for example, works below its optimum level when DT is reduced (to provide more output), whereas the accuracy of Euler's method remains the same;

3) situations under investigation display mild "stiffness". If a model contains both slow and fast acting dynamic elements, then it is described as "stiff". SD models can be considered "mildly stiff", if stable outputs from "stiff" equations (i.e., those that exhibit stiffness) are obtained when "non–stiff" methods are used for simulation, without having to utilise excessive computer processing time (i.e., needing to reduce DT to make the output stable). In these situations, simulation output will only be due to slow elements, as transients of fast elements die. Large DT decreases accuracy (one is unaware of the output from element interaction between time steps), with a likelihood of displayed output being unstable (fast and extreme transient output oscillations). This is known as "hunting"—the simulation model is attempting to maintain stability, but is unable to consider the important dynamically fast and slow elements together. To counter this problem a time step *must* be determined by the size of DT appropriate for fast, rather than slow elements.

As the DT time step (length 'h' in Figure 5.8) is reduced to take into account missing interactions, there is a requirement to use special "stiff" equations for integration. These are higher–order integration methods (for example Runge–Kutta Method) and require increasing amounts of computer processing time.

5.5.4 Runge–Kutta Method

This integration method can be used for solving "non–stiff" differential equations of the *n* th order. The "classical" method is of a fourth–order, but third and fifth are also widely used, (first–order being Euler's). The Runge–Kutta Method, as an alternative approximation, considers:

$$t(x_0) = t_0 \tag{5.26}$$

by computing:

$$\left.\begin{aligned} k_1 &= \mathrm{h}f(x_0 + t_0) \\ k_2 &= \mathrm{h}f(x_0 + \tfrac{1}{2}\mathrm{h},\ t_0 + \tfrac{1}{2}k_1) \\ k_3 &= \mathrm{h}f(x_0 + \tfrac{1}{2}\mathrm{h},\ t_0 + \tfrac{1}{2}k_2) \\ k_4 &= \mathrm{h}f(x_0 + \mathrm{h},\ t_0 + k_3) \end{aligned}\right\} \tag{5.27}$$

Then:

$$t(x_0 + \mathrm{h}) = t + \tfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \tag{5.28}$$

The Runge–Kutta method reduces error inherent in numerical approximation, especially Euler's Method. Error is in the order of h^5 and is considered fairly accurate for large values of 'h', (Knight and Adams, 1975). Replacement of differential Equation 5.20 by Equations 5.25 and 5.28 will be a compromise between accuracy and speed in computation. Reduction in time step—Delta Time (distance 'h' in Figure 5.6 i.e., DT or Δ), will also improve accuracy, but again increase computation time. This method, in general terms, sub divides the DT step, determining integration function 'h' based not only on the previous integral value, but also at the beginning of that step.

5.5.5 Problems With Integration Methods

Pugh, III (1980), when considering Euler's method with DT set at one–half the plotted intervals, found that an error ≤2.5% is produced, whereas when a fifth–order Runge–Kutta Method is used to four decimal places, there is no measurable error. However, computation through relevant Level, Rate and Auxiliary equations is three times higher. Additional errors can also occur when small time steps are used and hence large numbers of integrations. Such errors are caused by computational accuracy. Known as "round–off errors", these can be ten times higher for a fifth–order Runge–Kutta than Euler's method (Pugh, III, 1980: 81–2), unlike "global error" due to the choice of integration method. For example, if DT is set at 9.766×10^{-4} when solving:

$$x' = 1 - x \qquad \textbf{(5.29)}$$

$$y' = 1000(x - y) \qquad \textbf{(5.30)}$$

$$x(0) = y(0) = 0 \qquad \textbf{(5.31)}$$

Statistical methods can be used to assess effects of "round–off error", (Monroe, 1962). Such integration errors are out of the control of the modeller (unless he uses a different computer!) but can be reduced even though it is partially due, unlike an analog computer, to a digital computer's inability to perform continuous time integrations (Britting, 1976).

A compromise between the Runge–Kutta Method (with a high–order integration) and Euler's Method would be a reduction to a form of third–order integration. This however presents similar problems to one using a higher–order, in that functions within a model may not be continuous naturally and cause large integration errors. These are exacerbated by use of DYNAMO's CLIP and STEP (Pugh, III, 1983) functions for example (termination or commencement of a variable). Euler's Method, despite its simulation limitations, appears to be the best integration method for graphical outputs needing frequent outputs, for use in situations displaying occasional "mild stiffness" and where accuracy is not of prime importance.

Table 5.1 compares the accuracy between Euler's, second–order and forth–order Runge–Kutta Methods (*Stella for Business Manual, V2.1*: C–7).

5.6 USE OF DYNAMO FOR DECISION MAKING

A model is *not* a perfect representation of reality that can be trusted to make better decisions than decision makers. It is a flexible tool forcing decision makers to think harder, analyse their plans and policies before implementing them. This was discused earlier, but in summary, with respect to SD and use of DYNAMO:

> Rather than rely on intuition, an alternative procedure is to build a mathematical model of the situation [as perceived], and simulate its behaviour with a computer.... Any dynamic (changing over time) system...can be modelled with DYNAMO. (Pugh–Roberts, 1984: 2)

Table 5.1
Comparison of numerical errors in Euler's,
second–order and forth–order Runge–Kutta Methods

Time	Exact Value ($100 \cdot e^{-0.5 \cdot t}$)	Error Euler's Method (dt = 0.025)	Error 2nd–Order Runge–Kutta (dt = 0.05)	Error for 4th–Order Runge–Kutta (dt = 0.1)
0	100.000	0.0	0.0	0.0
0.1	95.123	0.0300	–0.000505	$-2.583e^{-7}$
0.2	90.484	0.0570	–0.000960	$-4.913e^{-7}$
0.3	86.071	0.0813	–0.00137	$-7.011e^{-7}$
0.4	81.873	0.103	–0.00174	$-8.892e^{-7}$
0.5	77.880	0.123	–0.00207	$-1.057e^{-6}$
0.6	74.082	0.140	–0.00236	$-1.207e^{-6}$
0.7	70.469	0.155	–0.00262	$-1.339e^{-6}$
0.8	67.032	0.169	–0.00285	$-1.456e^{-6}$
0.9	63.763	0.181	–0.00305	$-1.558e^{-6}$
1.0	60.653	0.191	–0.00322	$-1.647e^{-6}$

As a decision making model for frequent and general use, DYNAMO appears to be limited. Firstly, there is a requirement for a working knowledge of the computer language (although considerably lower than a standard high level programming language such as ‘C’ (in both standard and ++ object oriented formats) and standard Pascal. Secondly, there is no external facility (to the model) for data access—data being written into the original model and changed only by identification of appropriate equations (procedural knowledge).

For such a data–base system to be utilised there is a need for a software link between the application program (simulation model) and data–base (storage of data). This data–base management system (DBMS), together with a requirement to understand, for maintenance purposes, schemas and sub–schemas (Martin, 1976) would add further complications if DYNAMO were used. Creation of an efficient bespoke system would be time consuming as it is not possible to integrate a proprietary data–base system, for example Ashton Tate’s *dBase III plus*, due to incompatible operating systems, see Section 5.8.1.

There is no provision with DYNAMO for a user friendly interface with either an expert system or utilisation of a WIMP (Windows, Icons, Mice and Pulldowns) system, DYNAMO being menu driven. Work is currently being carried out by Khoshnevis and Austin (1986) and Gonzalez and Fernandez

(1986) in an attempt to reduce user unfriendliness. This work involves development of an intelligent natural language, human–to–computer interface. Thirdly, micro based DYNAMO (Micro–DYNAMO) is only capable of dealing with up to three hundred variables on both Apple II and IBM.PC/XT computers.

5.7 Hardware Limitations of Micro–DYNAMO

An IBM.PC system is restricted to a 640K byte RAM (Random Access Memory) limit addressed by the UCSD p–System IV.1 environment used to run Micro–DYNAMO, (rather than making use of standard DR/MS/PC.DOS (Disc Operating System) that can only address 640K RAM with pre Version 3 without getting into the complication of using/addressing extended and expanded memory with higher versions of DOS).

Micro–DYNAMO compiler is also limited to 8/16–bit (Intel 8086 and 8088 CPUs (Central Processing Unit)) technology excluding faster, true 16–bit (Intel 80286) and latest 32–bit (Intel 80386 and 80486) CPUs (see Appendix 3). Micro–DYNAMO is also unable to address an Intel 8087 numeric co–processor, which would speed up mathematical calculations. Some of these restrictions have been resolved by DYNAMO Professional which can be used under DR/MS/PC.DOS. This software is also not available for computers using any of the 32–bit Motorola 68000 series of CPUs.

Use of Micro–DYNAMO for reasons given above, limits the detail needed to produce a realistic LSP model simulation. Size of these model interrelations is such that important details would have to be restricted for the sake of maintaining a complete model structure (reducing resolution).

5.8 Limitations of SD Software

5.8.1 IBM.PC Based Micro–DYNAMO

If an accurate output is required, but the model exhibits little dynamic behaviour or "stiffness", there are other numerical methods better suited to model with. Integrated into this is DYNAMO's requirement for a modeller to select the DT time step to be used during the simulation period. The normal criterion is for 0.25 or 0.5 of the shortest model time constant ($T_n \rightarrow T_{n+1}$) to be used, but it is difficult to determine what this should be. In practical terms

DT is initially set at one (the largest Micro–DYNAMO will consider) and then undertake trial simulations with reduced time steps. Accuracy between runs is compared, until a compromise is reached between this and the time a computer takes to run through a simulation.

UCSD p–System software environment used in Micro–DYNAMO is self contained and limited to running specific Pascal programs. By using p–code (similar in appearance and level to an advanced assembly code) it is able to address a computer's Basic Input/Output System (BIOS) for file management together with a Pascal compiler and operating system. By writing a program in Pascal the inbuilt translator program is able to convert it into p–code, for execution by the compiler. Micro–DYNAMO needs to be recompiled on each simulation (but not simulation reruns with temporary data changes).

As SD simulation uses large information/detail "chunks" it is unable to take into account (absorb) smaller details, that would be otherwise needed for more complete analytical solutions. These simulations are able to provide general trends only, outputs from influences and occurrences being a result of the representation of RW structures and parameters. Starr (1980: 52) states that a SD format of Levels and Rates ignores '...algebraic conditions, discrete time formats, and random disturbances.' In response to this, analytical solutions are not required for the SDM, therefore, with respect to Equations 5.1 and 5.2, the behaviour characteristics of *A* and *B*, together with partial derivatives of functions *'f'*, using SD simulation (Micro–DYNAMO etc.), need not be ***exactly*** solved. Coddington and Levinson (1955) consider the determination of behaviour in advanced differential equations in terms of *A* and *F(x)* in:

$$\dot{x} = Ax + F(x) \qquad \textbf{(5.32)}$$

where:

A = linear elements;

$F(x)$ = nonlinear elements.

Starr (1980) drew together similarities and complementarity between SD simulations and analytical solutions. Burns (1977) attempted to convert a SD flow diagram (from digraphs or causal loop) into differential equations, by using set notation and a Square Ternary Matrix (STM). This could provide a link between a DYNAMO style format into a pure mathematical format able to cope with deficiencies found in the former. This idea still benefits a modeller as it involves other interested parties in defining system structure (see Section

5.10), without complications of needing to understand computer simulation mathematics.

5.8.2 Apple Macintosh Stella

Stella (Structural Thinking, Experiential Learning Laboratory with Animation) is an alternative SD software package, but only available to run on Apple Macintosh computers. An advantage of using this package, like most Macintosh computer software, is its use of a more friendly human/computer interface. This "graphics–orientated user interface" or GUI (gooey) utilises a WIMP system, making it easier but slower to use (there is a compromise between ease of use and speed in most computer systems).

Slowness is brought about by having to simplify model conceptualisation, due to Stella's limiting use of levels and delays, making structuring, in certain circumstances difficult. A "gooey" system is well suited to Desk Top Publishing (DTP) and the document presentation 'niche market' which Apple Macintosh now dominates. However, some of the computer's operating characteristics (such as use of a mouse and windows) can be utilised in any continuing development of PROMISS. This type of "user friendly" interface has been used as an enhancement to the standard PC operating system DR/MS/PC.DOS) in the form of Microsoft Windows, Version 3.

5.9 SD Methodology—Criticisms

SD was conceived as a quantitative means for dealing with dynamic and non–linear behaviour problems concerning management of social systems. Classical Operations Research (OR) provides an analytical tool kit for many situations, but is unable to deal with the complexity that SD can. OR considers decisions as being made within an open–loop process where process inputs are unaffected by the decisions themselves, (Forrester, 1968b; 1968c). SDM considers that within social systems where a boundary has been identified, any policy decision will be affected by others, i.e., future decisions will be a result of, and affected by, present and past ones, (see Section 5.5.1).

A major criticism of what appears to be the ability of people to conceptualise the complexity of social systems as multiple–loop, nonlinear feedback systems (Forrester, 1976). Ansoff and Slevin, (1968: 383) considers that SD:

> ...is not a well circumscribed body of theory, it is possible to define it through a series of stages of stages [sic] of increasing specificity. Therefore, the range of its application is dependent on the definition one adopts.

In this case, a definition refers to a particular situation rather than its application, as a theory of system structure and its associated dynamic behaviour.

In response to Ansoff and Slevin, Forrester (1968c) refutes criticisms concerning SD's theoretical foundations as "a body of theory". Although Forrester did *not* accept the critics' four point criteria for what constitutes a theory (1968c: 604 with reference to Ansoff and Slevin, 1968: 394), a point–by–point response was made. This being:

1. It should embrace a well defined body of observable variables.
2. It should have an explicitly stated set of hypotheses about these variables.
3. It should present a calculus, a method which, relying on the hypothesis, permits construction of statements about the variables.
4. Finally, in the very nature of the above definition, a theory is a statement of its own limitations, an implicit definition of areas of experience to which the theory does not apply.

Reference to the first point, the embracement of '...a well defined body of observable variables' could be considered as the major SD components of Rates and Levels, identified within the SoI boundary. The second point could be the structure of these components and feedback loops they eventually form.

The third point is answered from two directions. Computation of variables (which are hypotheses of a RWS—as is any model) is such, that their dynamic and interrelated behaviour is considered as a whole. Secondly, at the time these critical papers where written, there was a dearth of information concerning theory of dynamic behaviour within nonlinear systems. There was an implicit reliance on other theories incorporated within SD, for example the mathematics of feedback. This problem was (partially) resolved by Forrester's book *Principles of Systems* (1968a).

The fourth criterion was disregarded, as SD is not applicable to situations/problems that: 1) lack systematic interrelationships; 2) where the future is unaffected by the past (and present); and 3) where dynamic changes are of little interest.

Forrester's (1968c) own theoretical ideas are based at two levels. Firstly, there is a general theory of systems within SD, although it is also an amalgamation of several ideas, for example cybernetics and servomechanisms. The first, a theoretical level, has advantages due to the unrestricted nature of systemic research and analysis. Specific theories are needed to transfer '...insights and predictions from one observable situation to another.' (Ansoff and Slevin, 1968: 395). The second level, is one concerning SD modelling as being a: '...theory of structure and dynamic behaviour for a particular class of systems. In this sense, systems belong to the same class if they can be represented by the same structure.' (Forrester, 1968c: 606)

Understanding of model structure, i.e., number of Rates and Levels (order/integrations), gives one the ability of being able to portray two different RW situations that exhibit similar characteristics. Once dynamic behaviour and structure are understood for one situation, knowledge about this could be transferred and help in the analysis of another totally unrelated situation. Forrester (1968c: 606–7) quotes dynamic similarities between an employment–inventory system and a swinging pendulum. Herbert Simon (1969a) considers a "black box" representing the RWS as being modelled by a distinctly different one, but of both having the same output characteristics.

Once a conceptual model has been developed using digraphs/causal loop diagrams (Coyle, 1977; Goodman, 1974) with Rates and Levels identified, a computer code is written (Coyle, 1983). Pidd (1984), considers the usefulness of these diagrams as a link between verbal description of a SoI and computer coding in the form of conceptual equations.

> Both dynamists and managers profess that a significant benefit comes before the simulation during the flow chart modelling [causal loop diagrams].... This appears to have a therapeutic effect on the managers, forcing them to crystallize decision making processes and to order their thoughts according to a systematic information feedback model. (Ansoff and Slevin, 1968: 393)

DYNAMO modelling projects known variables by time steps determined by the model's DT (see Section 5.5.2), with the object of determining the system's future behaviour when: 1) variables known at time .K remain unchanged, or 2) when, due to policy changes one or more variables are changed. This will show or identify, in Equation 5.33 below, how model parameters (p) deviate from their nominal value or trajectory when $p \neq p_N$. Hearne (1987) describes SD equations as being in the form:

$$\frac{dx_i}{dt} = F_i(x, p, t) \qquad (5.33)$$

when:

$i \;=\; 1, 2, ..., n$

$x \;=\; (x_1, x_2, ..., x_n)$

$p \;=\; (p_1, p_2, ..., p_m)^T$

and:

$$l_i \leq p_i \leq u_i \quad \textbf{(5.34)}$$

when:

$i = 1, 2, ..., m$

l_i and u_i are known lower and upper bounds.

Hearne (1985; 1987) and Graham (1980) identified further limitations of DYNAMO in the area of parameter sensitivity. As in any real world SoI, many variables are stochastic in nature, rather than deterministic, therefore several computer scenarios need to be considered. Criticism was based on the need of such scenarios, changing one parameter at a time before running the simulation again. This limitation reduces the combination of changes found in the RW situation a model is trying to represent. Multiple parameter changes are discussed by Legasto and Maciariello (1980); Hearne (1985); Starr (1980) and parameter sensitivity involved in Forrester's World models by Vermeulen and Jongh (1976). When considering a model with 14 parameters and using a high, nominal and low value for each, there would be a need for 3^{14} combinations or 4.78 million simulations (this value is disputed in Section 7.6 where it should be 7.174 million simulations).

If P is a set of all m–dimensional vectors, with components meeting the criteria of Equation 5.34 for any $p \in P$ in Equation 5.33, $x(p,t)$ will be a path with $p = p_N$ i.e., trajectory of a nominal parameter vector, then $p_N \in P$ can be found (s (t)):

$$\sigma(t) = x(p_N, t) \quad \textbf{(5.35)}$$

To reduce the number of scenarios in Equation 5.33 one can let the sensitivity analysis problem become one of standard optimisation, by letting 'f' be the difference between the nominal trajectory and other options, these options having a difference interval from $t_0 \rightarrow t_f$. Maximizing $f(p)$ subject to Equation 5.33 the choices of 'f', with limits of $t_0 \rightarrow t_f$, will be:

$$f(p) = \frac{1}{n}\int_{t_0}^{t_f}\sum_{i=1}^{n}\left(\frac{x_i - \sigma_i}{\sigma_i}\right)dt \quad \textbf{(5.36)}$$

Here, interest is limited to a small number of variables, therefore in this equation it is appropriate to sum them only. But by the very nature of the interrelationships of each model variable, it would *not* be prudent to consider any one of them in isolation. A scenario *must* be carried out on them all.

Forrester (1970) identified model outputs as not being always predictable or initially having a logical (counterintuitive) leverage/influence, until simulation outputs have been obtained. For this reason, assumptions should *not* be made about significant/insignificant elements. The number of variable changes, as Hearne rightly points out, can be reduced by making use of existing knowledge i.e., the feasibility of some $p \in P$ is dismissed, because resulting solution trajectories will deviate from what is already known. This known data can be used to calibrate a model, using r components of $s_{i_j} = 1, 2, \ldots r$. The following equation is derived by using $t_0 \rightarrow t_1$ or DT; as the time interval:

$$f(p) = \int_{t_0}^{t_1} - \sum_{i_j=1}^{r} \left[\frac{\tau(x_{i_j} - \sigma_{i_j})}{\sigma_{i_j}} \right]^s dt + w \int_{t_1}^{t_f} \left(\frac{x_{i_j} - \sigma_{i_j}}{\sigma_{i_j}} \right)^2 dt \qquad \textbf{(5.37)}$$

where:

$$\sigma_{i_j}(t),\ t \in [t_0', t_1],\ i_j = 1, 2, \ldots, v: \text{when } r \leq n \qquad \textbf{(5.38)}$$

are the components of the nominal trajectory evaluated against known data;

$$\sigma_{i_j}(t),\ t \in [t_1', t_f],\ i_1 = 1, 2, \ldots, v: \text{when } v \leq n \qquad \textbf{(5.39)}$$

being components of a nominal trajectory of interest to an analysts/model builder. A specified parameter *'r'* is used to control how closely other trajectories follow the nominal criterion for: $t_s[t_0, t_1]$; if *'s'*, as an even integer, is made large enough, it will affect the first integral and hence restrict trajectories to within $(^1/_t)$ of the nominal trajectory, when $dt = [t_0, t_1]$.

It is noted that a weighting factor *'w'* has been introduced into Equation 5.37. If this inclusion is made then eventual results, as model outputs, cannot be value-free. This makes construction of any model, which is initially subjective —it being a construct resulting from the modeller's perception of the "real world", even more subjective.

Inclusion of these pre-determined weightings further complicates and multiplies scenarios needed to account for possible SoI variations. This complication is contrary to SD as a methodology, that of a situation being influenced by its structure, rather than data. Multiple scenarios are needed only to identify variable(s) with greatest leverage, (Forrester, 1986b). Data that will/can be changed as a policy change need only be used to initialise and change pre–simulation computer scenarios. Further to this, multiplication of simulations are increased as 'f' is changed. Analysis of parameter sensitivity problems in SD modelling does not reduce the number of simulations, but increases them.

Starr (1980: 54) utilises nominal values by using an expanded solution in a Taylor's Series, together with the use of Tomovic and Vukobratovic's (1972) analytical sensitivity analysis. Taking a system represented by the nonlinear Equation 5.2, denoting its solution by:

$$\phi(x_0, a, t)$$

and then defining:

$$DXA = \frac{\partial x}{\partial a} \quad , \quad n \times p \text{ matrix of parameter sensitivity coefficients.}$$

Elements of the *DXA* will be time dependent and as such the behaviour can be defined by partially differentiating:

$$\frac{\mathrm{d}}{\mathrm{d}t}(DXA) = \frac{\partial f}{\partial x}(DXA) + \frac{\partial f}{\partial a} \tag{5.40}$$

From trajectory $f(x_0, a, t)$, indicated partial derivatives can then be evaluated. These equations appear to be linear in *DXA* with characteristics of growth, decay etc. being related to the matrix $\partial f/\partial x$. Difficulty has been found using this representation in determining parameters/elements with greatest leverage, although McClamroch (1976) carried out a study to correct this shortfall.

McClamroch (1976) attempted to design a technique for determining links between system stability and element relationships. By developing a series of theorems for analysis of, and use with, nonlinear multivariable systems, situation/method stability can be determined. This is carried out on a loop–by–loop (considering each as a linear subsystem) basis using a scalar circle condition, thereby determining critical elements that cause system stability.

This study, like Starr's, attempts to view a system in an analytical linear format, to find elements of greatest leverage. Such studies have been undertaken because of the difficulty in identifying such individual elements in situations conceptualised in a nonlinear format. Starr (1980: 55) considers the idea of making changes in groups, but unfortunately this: '...would introduce an overwhelming number of combinations and hence are never done on an all inclusive basis.' This is demonstrated by using a second–order linear equation to show problems in changing groups of elements/parameters, rather than on an individual basis:

$$a\ddot{x} + b\dot{x} + cx = 0 \qquad \textbf{(5.41)}$$

where: *'a'*, *'b'*, *'c'*, are representative of system parameters (a logic function in which selection of *'a'* or *'b'* is dependent upon the value of *'c'*). Rather than consider these individually, as one would in a normal SD simulation, they are described by the natural frequency:

$$\omega = \left(\frac{c}{a}\right)^{\frac{1}{2}} \text{ and the damping coefficient } \rho = \left(\frac{b}{2}\right)(ac)^{-\frac{1}{2}}.$$

It is considered that by analysing individual elements, parameters *'ω'* and *'ρ'* would be difficult to determine: 'The point is that behavior descriptors depend upon functions of parameters and linear theory and analytical formats can suggest forms of these functions.' (Starr, 1980: 56)

These identified deficiencies highlight further limitations in making DYNAMO an everyday computer simulation language for decision making purposes; it being considered only suitable as a general purpose language, where finer and exact output detail is not required. DYNAMO does, however, have the ability to quickly model a SoI holistically, using significant elements/variables only (although this is subjective).

When considering a decision making computer simulation model, more detail than that provided by DYNAMO is required, with an ability to introduce variable changes whilst the model is running (interactive simulation). DYNAMO, needing to be recompiled and simulated for the full model length, after each parameter change, does not meet this criterion.

In conclusion, SD is concerned with formulating general patterns and interrelationships (cause and effect chains) into models in a form of Rates and Levels. From such models, evaluation of policy scenarios can be undertaken, to determine for example, element(s) or parameter(s) with greatest leverage. It is for determination of answers to "what if" scenarios that these computer models should be used, ***not*** for accurate forecasting. This point appears to be missed by many critics, for example Ansoff and Slevin (1968). Rothkopf (1973: 60–1) offers a rather extreme lack of "confidence" in SD models in that they:

> ...should be viewed as no more than the sincere opinions of their builders. They should not be presented as scientifically justified conclusions. If they are, decision–makers may place excessive trust in them—and this may lead to bad decisions.

There can be some sympathy with this statement, but no model(s) should be presented, without direct involvement of the problem owner(s) and decision makers who will eventually use them. Validation of SD simulation models is discussed below.

5.10 Validation and Problems of Making DYNAMO/SD Models Representative

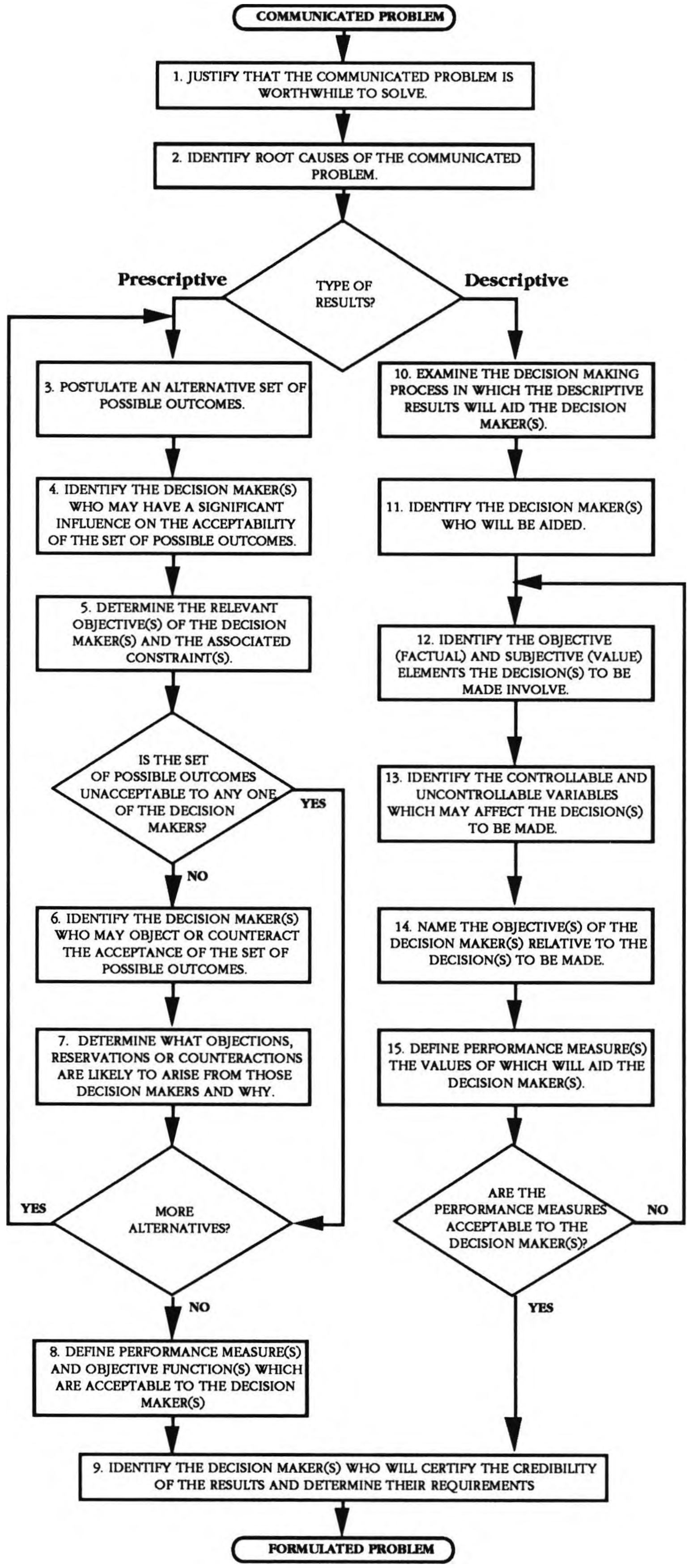
Simulation models are representations of the RW concerning both their static and dynamic characteristics (O'Keefe, 1986b), together with the modeller's perception of structure and process. This problem initially arises in the Problem Definition stage of the methodology. Balci and Nance (1985: 78) developed a flow chart for dealing with verification of the transformation of a RW situation, into a Conceptual Model, taking into account Descriptive/Prescriptive analysis, see Figure 5.9.

Extensive and concentrated effort over time is needed to tune and validate a simulation model and make it representative of the RW, to the decision maker's/modeller's requirements. Problems, such as these, bring about statements as: '...models are to be used but not believed.' from Theil (1971: vi) and:

> Scientific philosophy...refuses to accept any knowledge of the physical world as absolutely certain. Neither the individual occurrences, nor the laws controlling them, can be stated with certainty is attainable; but these principles are analytic and empty. Certainty is inseparable from emptiness; there is no synthetic ***a priori***. (Reichenbach, 1951: 23)

Resolving the validity problem is achieved by considering computer models in two ways: firstly, as being suitable for the purpose and problem environment being analysed; and secondly, being consistent with the RW it tries to represent. Shannon (1981: 574) proposed that three questions should be asked of when validating a model:

1. Does the model adequately represent the real world system?



High–Level procedure for formulating a problem
Figure 5.9

Table 5.2
Opportunities for errors in simulation modelling

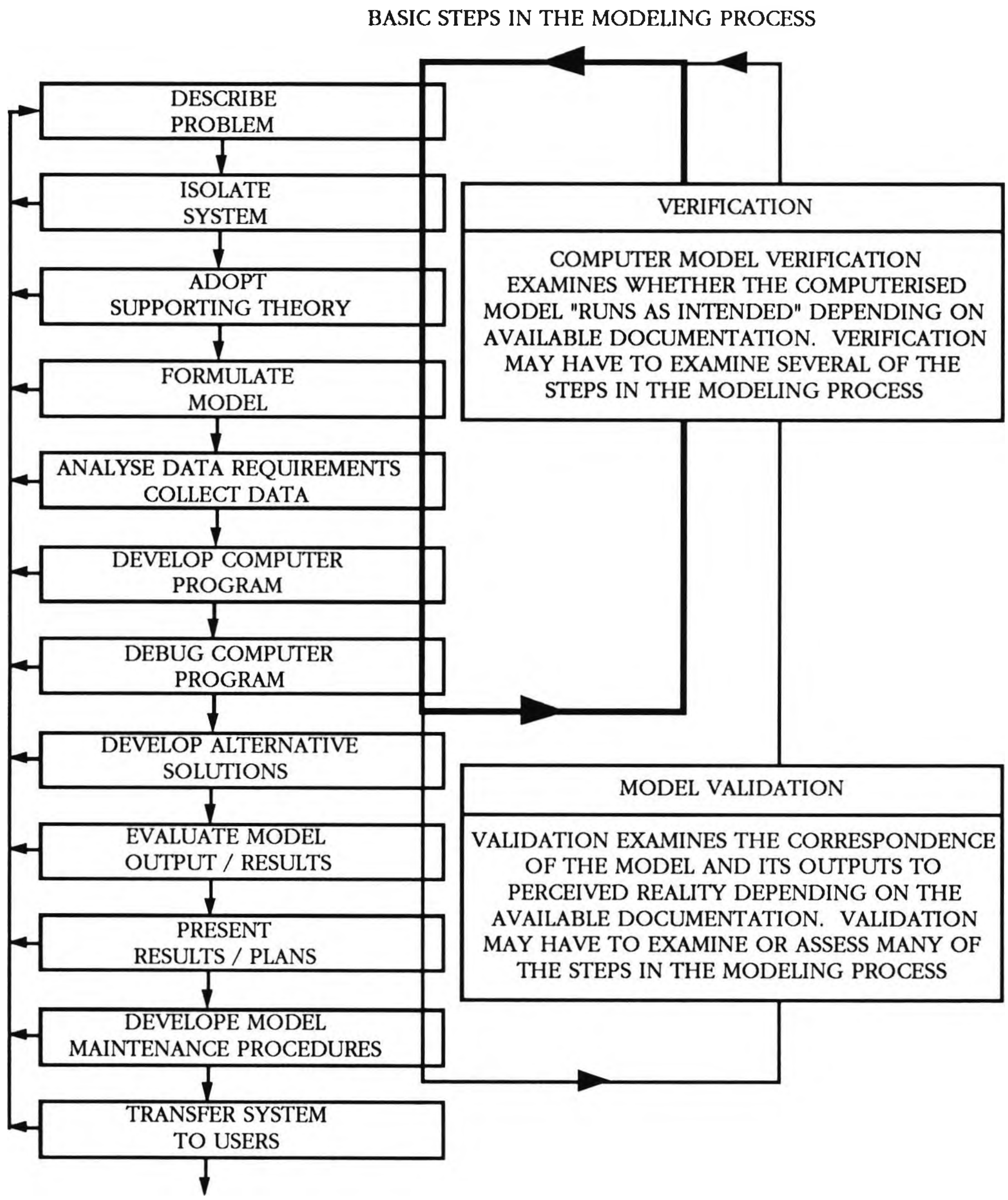
ELEMENT	POSSIBLE ERRORS
1 **REAL SYSTEM** The part of reality to be modelled;	**a)** Defining the system; **b)** Defining the boundaries.
2 **CONCEPTUAL MODEL** Model builder's perception and understanding of the RW situation;	**a)** Misunderstanding how the system works; **b)** Excluding relevant variables.
3 **EXPERIMENTAL DOMAIN** Description of the limited set of conditions under which the RW situation will be modelled;	**a)** Specifying goals of study.
4 **FORMAL MODEL** The construction of a relatively simple model that is valid in the experimental environment that has been chosen;	**a)** Design of model; **b)** Data used.
5 **COMPUTER MODEL** The step–by–step implementation of the formal model into computer coding;	**a)** Logic and coding.
6 **EXPERIMENTATION** The using of the computer model to generate behavioural data through planned experimentation and analysing generated data.	**a)** Procedure for use of model; **b)** Interpretation.

2. Is the model generating behavioural data characteristic of the real system behavioural data?
3. Does the simulation model user have confidence in the model's results?

He then considers the "Opportunities for Errors" that occur during the development of a simulation model, see Table 5.2.

Figure 5.10, Banks *et al.*, (1987: 14), shows a twelve stage model evaluation cycle. The first seven are used for verification and the first eleven validation, before it is suitable for a user. The former forms part of the iterative nature of the SDM as a formal process. The latter is necessary if a subsequent model is to be used with confidence, therefore it must be inherent. A model will be dynamic and used in a dynamic environment so it must be adapted as the RW situation changes, taking into account points identified in Table 5.2.

Forrester (1968c: 615), when referring to models as representations of the RW, considers "point of proof"—one can never 'prove' a model is an 'exact



Verification and validation flow chart
Figure 5.10

representation', therefore models should not be judged on an absolute scale for not being perfect. If models increase our ability to understand and be able to clarify available knowledge, then they should be judged on this ability alone: 'Model validity is a relative matter. The usefulness of a mathematical simulation model should be judged in comparison with the mental image or other abstract model which would be used instead.' (Forrester, 1968a: 3–4)

> Simulation models are normally resorted to when other forms of analysis will not usefully unravel the problem...but because of the large number of possible interactions, when combined, make it impossible to understand the behaviour of the total system. (McKenney, 1967: B–102)

Richardson and Pugh (1981: 311) also consider validation of SD computer models:

> In the system dynamics approach validation is an on–going mix of activities embedded throughout the iterative model–building process. No single test suffices to validate a system dynamics model or provide a measure of its degree of validity. An observer wishing to make a judgement about validity of a system dynamics study must follow much the same path as the modeller. There is no royal (or Student's) road to validation in the system dynamics approach.

Karl Popper (1959) suggests that models should be to some degree *confirmed*, rather than *verified*. If in a series of empirical tests of a model no negative results are found, but a number of positive instances increases, then confidence is increased in a step–by–step manner, as Carnap, (1963) states: 'Thus, instead of verification, we may speak of gradually increasing confirmation of the law'. 'If error is corrected whenever it is recognised as such, the path of error is the path of truth.' (Reichenbach, 1951: 326)

Validation of a model is the determination of truth, i.e., the means of proving a model to be true. This implies that one is aware of a criterion for differentiating between models that are true and those that are not and also have the ability to apply such criterion (Naylor and Finger, 1967). Validity of a model is made probable, not certain, by assumptions underlying the model; the inductive inference must be conceived as an operation belonging in the calculus of probability, (Reichenbach, 1951).

Forrester and Senge (1980: 209) consider that SD models should have confidence built into them, or '...accumulates gradually as the model passes more tests and as new parts of correspondence between the model and empirical reality are identified.' Table 5.3 shows a summary for building confidence into SD computer simulation models.

Here the focus is on structure and behaviour when testing for suitability and consistency. Core tests are essential if confidence is to be built into such models. Other tests may not be cost effective or even possible, but if they are carried out then confidence will be further enhanced.

Another problem with validating a computer simulation model is the language medium used. Many clients (problem owners, decision makers etc.) are not familiar with computers, software languages or even interpretation of outputs from such simulation models:

Table 5.3
Summary table of tests for building confidence
into System Dynamics computer simulation models

	Focusing on **STRUCTURE**	Focusing on **BEHAVIOUR**
Testing **SUITABILITY** for purposes (tests focusing inward on the model)	Dimensional consistency **(a)** Extreme conditions in equations **(a)** Boundary adequacy – important variables – policy levers	Parameter (in)sensitivity **(a)** – behaviour characteristics – policy conclusions Structural (in)sensitivity **(a)** – behaviour characteristics – policy conclusions
Testing **CONSISTENCY** with reality (tests comparing the model with information about the real system)	Face validity – rates and levels – information feedback – delays Parameter values **(a)** – conceptual fit – numerical fit	Replication of reference modes (boundary adequacy for behaviour) **(a)** – problem behaviour – past policies – anticipated behaviour Surprise behaviour Extreme condition simulations Statistical tests – time series analysis – correlation & regression
Contributing to the **UTILITY & EFFECTIVENESS** of a suitable, consistent model	Appropriateness of model characteristics for audience – size – simplicity/complexity – aggregation/detail	Counter-intuitive behaviour – exhibited by model – made intuitive by model-based analysis Generation of insights

(a) → Core Tests

> ...produces output resembling those observed in the real world, and inspires confidence that the real causal process has been accurately represented. However, because the assumptions incorporated in the model are complex and their mutual interdependencies are obscure, the simulation program is no easier to understand than the real process was. (Dutton and Starbuck, 1974: 4)

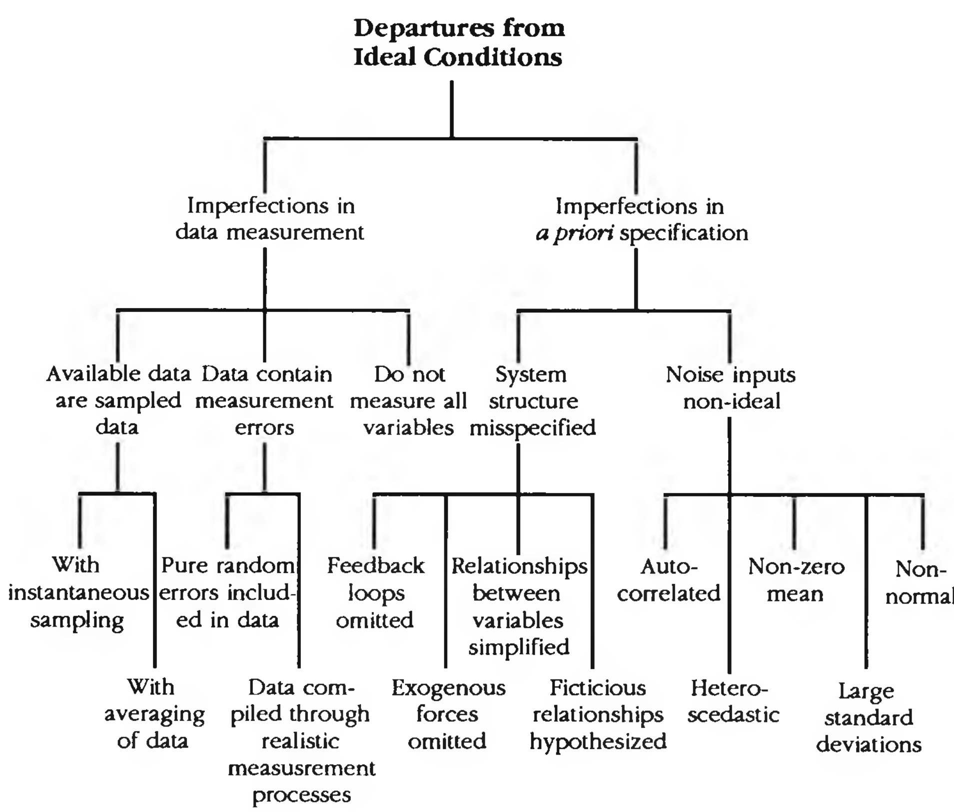
These points raise doubts concerning the soundness of Forrester's policy for validating SD models. Statistical methods are rejected for model validation by any other criterion than the extent of its usefulness, (Forrester, 1961; Schrank and Holt, 1967):

> ...propose that the criterion of the *usefulness* of the model be adopted as the key to its validation, thereby shifting the emphasis from a conception of its abstract truth or falsity to the question whether errors in the model render it too weak to serve the intended purpose. Schrank and Holt (1967: B–105)

'...the techniques commonly used by social scientists to analyze statistical data can be inconclusive or misleading when applied to the kind of nonlinear dynamic systems found in real life.' (Forrester, 1976: 31) Validation of computer simulation models and data generated are sampling rules resting entirely on the theory of probability, (Naylor and Finger, 1967).

Added to this is a problem of knowing whether or not available data is itself valid. Sargent (1984: 118) considers that little can be done about ensuring that such data is correct:

> The best that one can do is to develop good procedures for collecting data; test the collected data using such techniques as internal consistency checks and screening for outlets and determine if any outliers found are correct and develop good procedures to properly maintain and collect data.



Variations in experimental conditions
Figure 5.11

Senge (1977: 179) also highlighted problems of imperfect data caused through sampling being compiled on only a periodic bases, for example monthly or quarterly. These '...departure[s] from idea conditions....' are shown in Figure 5.11.

Unless a model can be validated, to both a model builder and prospective user's satisfaction, an error by these parties known as the "Halo Effect", can be made. When a computer model gives an indication of what is required, further examination is unlikely to the same degree then if it gave an indication one was not expecting. A simple model may provide less information over a long period, but its overall performance and hence validity, may be better than a more complicated one, (Crissey, 1974).

This problem is brought about by the determination of what is ***complex*** and what is ***complicated***. Complexity refers to substantive logic—mathematical interrelations and difficulties; complication can arise in almost any arrangement of facts, concepts, or thoughts. Complication is an undesirable characteristic of any construct; complexity may be an inherent feature. Complication often expresses lack of effort to give model construction its appropriate form, (Brewer, 1973: 4). An SD model (like any model) should help understand complexity:

> Time after time, one sees kinds of behavior in a system model the significance of which had previously gone unrecognized in the real–life situation. When, however, such behavior is recognized in the model one finds in the real–life system clear indication that the same dynamic modes exist in reality as well as in the model. (Forrester: 1968c: 607)

In such a situation it would be difficult to validate a model by any other means than put forward by, for example, Forrester. Validity of simulation models are difficult to ascertain through empirical observation if it demonstrates counterintuitive behaviour. If we do not understand a RW situation, thereby requiring a model to help, it is difficult to say it is *not* valid if the situation has been clarified because of it. One needs a model (whether consciously or not) to extract an understanding of the complexity, otherwise it is likely to remain a puzzle. It is therefore fair to confirm Forrester's point, that if a model does what it is conceptualised and designed for, i.e., to increase understanding, then it should also be considered valid:

> There is no method for proving a model to be correct. Einstein's theory of relativity has not been proven correct; it stands because it has not been disproven, and because there is shared confidence in its usefulness. (Forrester and Senge, 1980: 211)

It should be noted in conclusion to this brief discourse, that only a limited number of aspects have been considered, namely those put forward by proponents of SD. Leaning, 1980; Flood and Carson, 1988 represent other views.

5.11 CONCLUSION

This chapter has considered the specific systemic methodology of System Dynamics. SD is more widely known for computer simulation models than the means of their development. Consideration has been given to both aspects (see also Chapter 4 of Flood and Jackson, 1991). The former is based on system state indicators known as Levels and system flows determined by Rates. Methodological stages in model construction follow a general means of development from Problem Definition through to Evaluation—all being iterative with previous stages.

The later part of this chapter considered various debates and criticisms related to the mathematical aspect of computer simulations. Philosophical criticisms have also been included. In the past these have mainly been responded to by the 'Father' of SD, Professor Jay Forrester at MIT. Only the major papers have been discussed here. Such debates and resolutions still continue, but they do not have any effect on the use of SD, as a methodology, for developing a project computer simulation model. This is undertaken in the following chapter.

CHAPTER 6

Modelling a Project

> Persecution is used in theology, not in arithmetic, because in arithmetic there is knowledge, but in theology there is only opinion. So whenever you find yourself getting angry about a difference of opinion, be on your guard; you will probably find, on examination, that your belief is getting beyond what the evidence warrants.
>
> Bertrand Russell,
> *Unpopular Essays*, 1950

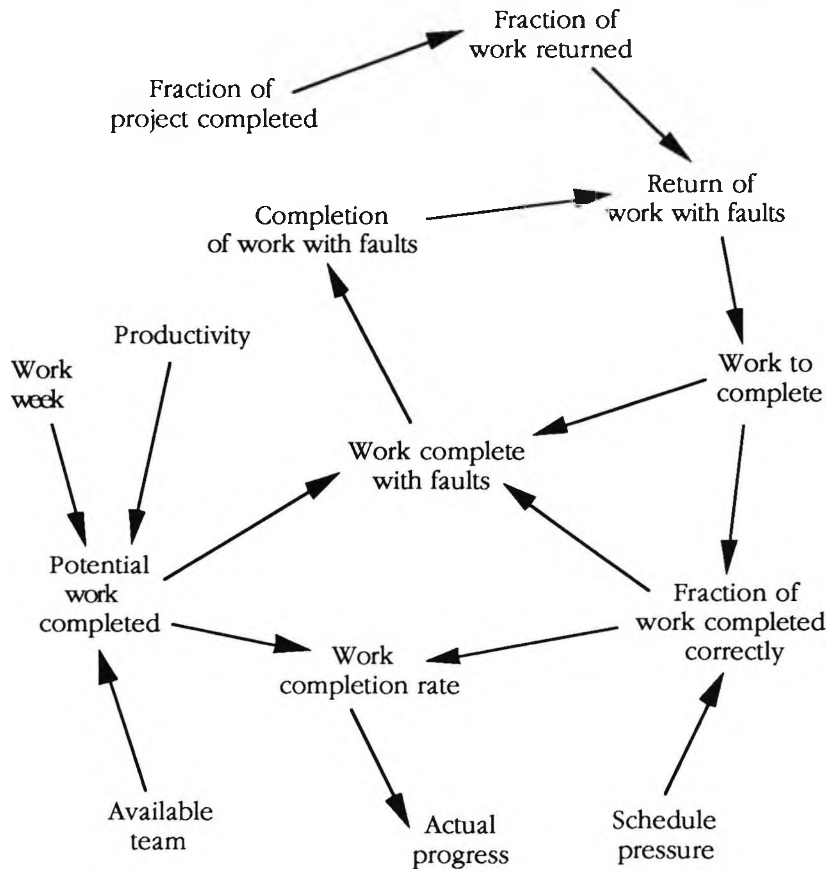
6.1 INTRODUCTION

Development of a mathematical computer simulation model for PROMISS takes into account a number of points discussed in previous chapters. Basic formulation is a hybrid of ideas and concepts developed by: Gordon, 1982; Richardson and Pugh, 1981; Richmond *et al.*, 1987; Stevens, 1986; 1987 (see Appendix 2); Stevens and Flood, 1988.

This chapter breaks down the PROMISS simulation model into various elements (variables) and briefly discusses them. Some of these points are subjective, but the basic principles are sound. PROMISS is made up of over six hundred variables (excluding program variables). The basic structure of PROMISS can, however, be broken down into just over one hundred strategic elements. Major concepts are discussed here, including computer listings (specifically Mechanical Fitters—MF). For PROMISS to be made for commercial use (in any project) elements should only be predetermined by the user. Stella (icon and Apple Macintosh based) and DYNAMO (equation and PC based) provide facilities for such a user defined input.

6.2 WORK PROGRESS

For a project to make progress, work being carried out must be completed without faults. If faults exist they will only become apparent some time in the future, when for example, completed work is tested. Figure 6.1 shows how work is completed, making actual progress.



Simulation model concept, Part 1
Figure 6.1

Work to complete is initially a quantity of activities (measured in man–hours for the duration of the CP) making up a project. During project completion this will be decreased as work is finished, but increase as work is returned with faults needing correction. As a project nears completion and testing intensifies so the fraction of work returned with faults increases (becomes known).

Potential work completed is determined by productive hours worked per week (productivity is discussed in Section 6.4.2) and size of available team (discussed in Section 6.5). These collectively influence work completion rate and hence actual progress. As potential work completed increases, so work completed with faults increases (as a proportion). The fraction of work completed correctly is also influenced by schedule pressure (discussed in Section 6.3).

6.2.1 Work Progress—Computer Listing

```
Actual_Progress_MF := Actual_Progress_MF  + T0 * (Work_
Completion_MF);
```

```
Compl_Work_wFaults_MF :=  Compl_Work_wFaults_MF + T0 *
(Work_Comp_wFaults_MF - Return_of_work_wFaults_MF);
Total_Rework_MF := Total_Rework_MF + T0 * (Incr_Total_
Rework_MF);
Work_To_Complete_MF := Work_To_Complete_MF + T0 * (-Work_
Completion_MF
Work_Comp_wFaults_MF + Return_of_Work_wFaults_MF);
Work_Week_MF := Work_Week_MF + T0 * (Chg_Work_Week_MF -
Puls_to_End_Project_MF);
```

(Ref. File SD.PAS 539, 542, 549–551)

Tables 6, 11 and 16 ([x, 1 → 14]) provide non–changeable interactions to control: fraction of work returned against fraction of project completed (6); fraction of work completed taking into account schedule pressure (11); and fraction of work with faults when compared with fraction of project completed (16).

There is a difference between Tables 6 and 16: Table 6 identifies when faults are detected; and Table 16 when those faults occur, both with respect to fraction of project completed.

```
{ Table function: _Fract_Faults_Detected_MF = graph(Fract_
Proj_Complete_MF) }
T[6,1]    := 0.000;  T[6,2]    := 1.000;  T[6,3]    := 0.100;
T[6,4]    := 0.000;  T[6,5]    := 0.000;  T[6,6]    := 0.020;
T[6,7]    := 0.060;  T[6,8]    := 0.100;  T[6,9]    := 0.190;
T[6,10]   := 0.305;  T[6,11]   := 0.480;  T[6,12]   := 0.880;
T[6,13]   := 0.975;  T[6,14]   := 1.000;
```

(Ref. File SDINT.PAS 268–273)

```
{ Table function: _Fract_Compl_Correct_MF = graph(Schedule_
Pressure_MF) }
T[11,1]   := 0.000;  T[11,2]   := 2.000;  T[11,3]   := 0.200;
T[11,4]   := 0.985;  T[11,5]   := 0.970;  T[11,6]   := 0.945;
T[11,7]   := 0.905;  T[11,8]   := 0.850;  T[11,9]   := 0.750;
T[11,10]  := 0.645;  T[11,11]  := 0.460;  T[11,12]  := 0.350;
T[11,13]  := 0.280;  T[11,14]  := 0.265;
```

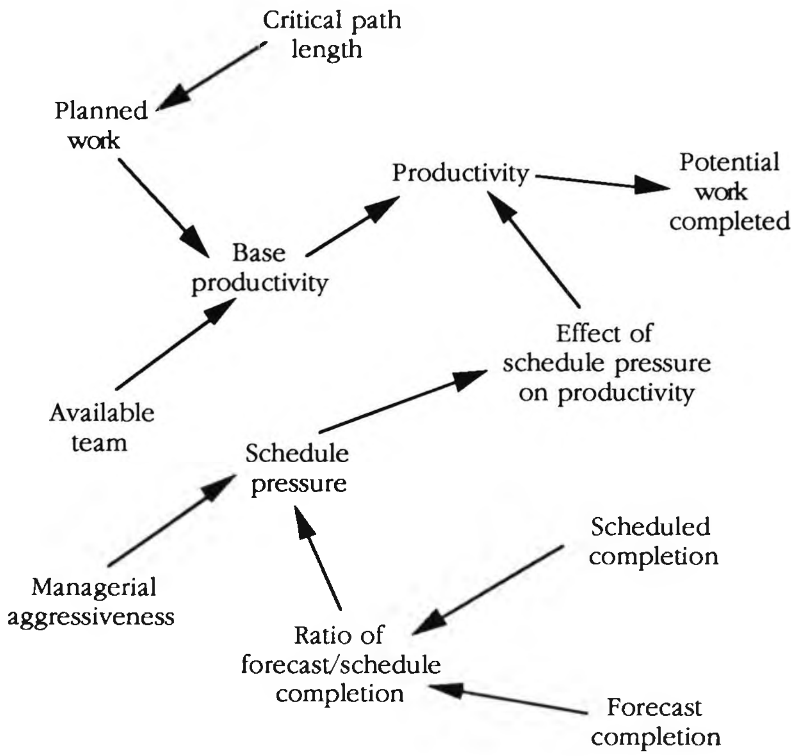
(Ref. File SDINT.PAS 303–308)

```
{ Table function: _Fract_Work_Return_MF = graph(Fract_Proj_
Complete_MF) }
T[16,1]   := 0.000;  T[16,2]   := 1.000;  T[16,3]   := 0.100;
T[16,4]   := 0.000;  T[16,5]   := 0.000;  T[16,6]   := 0.005;
T[16,7]   := 0.014;  T[16,8]   := 0.026;  T[16,9]   := 0.041;
T[16,10]  := 0.061;  T[16,11]  := 0.092;  T[16,12]  := 0.130;
T[16,13]  := 0.182;  T[16,14]  := 0.200;
```

(Ref. File SDINT.PAS 338–342)

```
Fract_Proj_Complete_MF := Actual_Progress_MF /
InitialProject_Work_MF;
TableWithHiLoCapability(Fract_Proj_Complete_MF, _Fract_
Faults_Detected_MF, 6);
Known_Rework_MF := Compl_Work_wFaults_MF * _Fract_Faults_
Detected_MF;
```

(Ref. File AUXRATES.PAS 128–130)



Simulation model concept, Part 2
Figure 6.2

```
Reported_Fract_Comp_MF := 1 - (Perceived_Work_Rema_MF /
InitialProject_Work_MF);
TableWithHiLoCapability(Fract_Proj_Complete_MF, _Fract_
Work_Return_MF, 16);
Return_of_Work_wFaults_MF := Compl_Work_wFaults_MF *
_Fract_Work_Return_MF;
Rework_as_Percent_MF := (Total_Rework_MF / InitialProject_
Work_MF) * 100;
```

(Ref. File AUXRATES.PAS 205–210)

```
Work_Completion_MF := (Potential_Work_Comp_MF * _Fract_
Compl_Correct_MF) / 5;
```

(Ref. File AUXRATES.PAS 217)

6.3 Project Work

Completion of a project is determined by actual progress. PMs however require to know, at any given point during a project, given past and current productivity rates, a forecast for completion date/time (see Section 6.4). When forecast and scheduled completion are compared, a completion ratio can be determined, (see Figure 6.2). Such a ratio creates schedule pressure. If forecast completion is to the left of schedule (completion will be early), completion ratio will be positive, but if to the right (completion will be late) negative.

Causes of an increase/decrease in schedule pressure by PMs is directly related to actual and planned (forecast) progress i.e., schedule slippage. Project length is determined by CP(s), therefore in reality this will cause managerial direction and resources to be focused to ensure minimal slippage. However, other parallel and non-critical tasks (those with float/slack) especially when LRs are limited, tend to be less well resourced and directed. Figure 1.4 showed a PTG identifying such a situation where non-critical paths have become critical and CPs super–critical. Once this has been identified and acknowledged, additional resources and managerial direction need to be applied to what are now new CP(s)—hence increasing managerial pressure.

Planned work at each time step along a CP will be derived from either levelled (for each LR priority) or original PERT unlevelled data file—see Section 3.4 and Figure 3.1. By combining this with the available teams, the core function of PROMISS, with respect to LRAs, can be found. Planned work provides the model with information concerning the quantity of work to be completed and when it is required. Productivity becomes a critical element, described further in Section 6.4, as it provides data to determine potential work completed, a variable determining what is likely to be achieved.

6.4 Actual Work Completion

Actual progress determines the fraction of project completed when compared against initial project work. The reported fraction of work completed will need to take into account how much work is completed as an advancement towards project completion. As some work is reported completed, it will contain faults—known rework needing rectification.

Forecast completion and schedule extension influences changes in schedule, see Figure 6.2. This in turn determines scheduled completion. By definition, a LSP is probably "one-off" and it is difficult to be sure, with respect to progress, where one should be at any given point. Experience has shown that PTGs produced by a PERT provide an important indication of how a project is progressing. The importance of such information is at it strongest during the middle stages of the project. During the early stages there is a greater time tolerance to rectify slippage and in the later stages a clearer picture is apparent as the amount of work remaining, against available time and resources, is more readily identifiable. It is at this stage that managerial pressure has greatest effect. When PMs are in receipt of actual productivity information, making predictions of a realistic completion date can be made. Labour resources deployed are also then likely to be more receptive to pressure from PMs to complete at a higher productivity rate. As schedules slip, so managerial pressure should be increased.

Once a LSP contract has been started it is assumed that all work to be carried out has been identified, with key dates agreed, see Figure 1.6. Additional work (other than rectification work) must be sanctioned by the customer and PM. Provided that such an extension does not occur, then project progress will be monitored against the original planned completion date. It is necessary for PMs to be able to forecast, from the current completion rate, what a likely completion date will be. A positive discrepancy (i.e. one where forecast is to the right of planned i.e., schedule slippage) will increase managerial pressure.

In many circumstances there is a need to increase productivity (for example when projects near completion) over and above the normal achievable by LRs. However, changes in pressure can have both positive and negative effects, for example if pressure is constant but applied at the wrong time (too early or too late). Alternatively, pressure is inconsistent with a situation. An example of this latter point would be application of excessive pressure early on in the project, when it was not really necessary (as perceived by those LRs to which pressure is being applied)—the result being that when pressure is really needed, there would be little or no effect. The amount and application of such pressure is subjective and totally dependent upon the individual situations managers find themselves in. The effect of pressure outside these situations can be modelled—it is assumed that some form of pressure will increase productivity, to some degree, for a period of time. After that, pressure will be counter productive.

6.4.1 Actual Work Completion—Computer Listing

```
Scheduled_Completio_MF := Scheduled_Completio_MF + T0 *
(Change_in_Schedule_MF);
```

(Ref. File SD.PAS 547)

Table 1 shows how schedule pressure can affect productivity.

```
{ Table function: _Effect_of_SP_Prod_MF = graph(Schedule_
Pressure_MF) }
T[1,1]   := 0.000;  T[1,2]    := 2.000;  T[1,3]    := 0.200;
T[1,4]   := 0.480;  T[1,5]    := 0.510;  T[1,6]    := 0.600;
T[1,7]   := 0.740;  T[1,8]    := 0.860;  T[1,9]    := 1.000;
T[1,10]  := 1.110;  T[1,11]   := 1.200;  T[1,12]   := 1.290;
T[1,13]  := 1.340;  T[1,14]   := 1.350;
```

(Ref. File SDINT.PAS 233–238)

```
  Base_Productivity_MF := Planned_Work_MF/Week_Hrs/
  Available_Team_MF;
if Base_productivity_MF > Max_Base_Prod_MF then
  Max_Base_Prod_MF := Base_productivity_MF;
If (TIME div 5) >= Crit_Path_Length then
  Base_Productivity_MF := Max_Base_Prod_MF;
```

(Ref. File AUXRATES.PAS 124–128)

```
Schedule_Pressure_MF := 1 + ((Ratio_Forcst_Sched_MF) - 1)
* Mngrs_Aggressivenes;
```

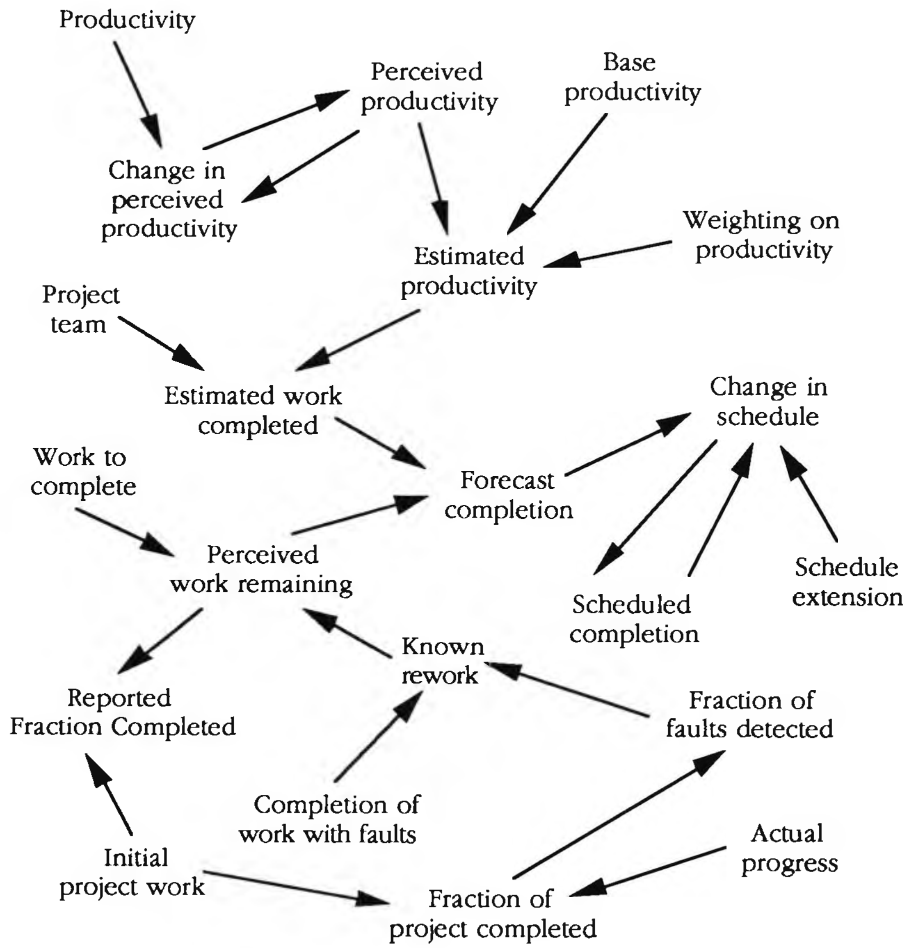
(Ref. File AUXRATES.PAS 149)

```
Schedule_Slip_MF := ((Completion_Date_MF-Target_Compl_
Date_MF) / Target_Compl_Date_MF)*100;
```

(Ref. File AUXRATES.PAS 212)

6.4.2 PRODUCTIVITY

Completion of a project is determined by the productivity of LRs deployed. There are two types of productivity: base and perceived, leading to formulation of estimated productivity, see Figure 6.3. Estimated productivity provides an estimate of work completed and therefore forecast completion.



Simulation model concept, Part 3
Figure 6.3

Estimated productivity, when analysed by the project team, provides information concerning estimated work completed and hence forecast completion. Such productivity is the difference between base productivity and perceived productivity. Initial estimates of productivity are compared with actual productivity so PMs can estimate future productivity. Therefore, during early stages of a project, greater weighting needs to be placed on base productivity. This in turn allows for a more realistic completion date to be determined.

Whilst determining how long each activity will take, when compared with times estimated by a trade estimator, productivity of LRs working need to be known. This value can be predetermined from historical records, for example Mechanical Fitters may have a rate of 0.7 and Painters 0.75. This implies that initial estimates for each activity need to take into account average known productivity, but *not* changes as a result of environmental influences which are unknown.

PROMISS calculates base productivity as estimated work over time (in hours) with reference to the number of available LRs. From this figure, which varies as the planned work pattern changes (as dictated by PERT network schedule), productivity is determined. Productivity takes into account the effect of pressure from changes in managerial direction and schedule pressure and thus influences the amount of potential work accomplished.

Base productivity is determined, in general terms, by LR experience and quality (most estimates take this as being an average person, adequately trained, having access to necessary resources to accomplish a task). The quality of LRs can be enhanced by increasing training. This, although important, is difficult to take into account when estimating how long each activity will take. Those making such estimates/calculations are unlikely to know which LRs will be undertaking any given task, therefore, only averages can be used. This base productivity figure is used for each type of LR, as a nominal criterion.

Another factor also to be taken into account when defining base productivity, is that known (historical) productivity may not be achieved during a new LSP. Only whilst a project is underway can PMs know how realistic that original productivity estimate was. Even with this knowledge, it can still be difficult to predict what future productivity is likely to be. Environmental changes, for example, in a LSP, can influence attitudes of individual LRs. Many of these factors are change factors determined, in part, by LMs.

Using this criterion, two factors: adequate training; and access to necessary resources, can be directly influenced by managers (both line and project). The first factor is subjective, as productivity improvement can only be increased when training is coupled with experience—training provides the key to open a door and experience the ability to open it. Therefore, any improvement as a result of training, can be difficult to predict over the length of most LSPs. The second factor can be a result of good management (and should be part of a good working environment). Unfortunately in practice this is difficult to achieve, especially in "one–off projects" that may require unique tools and facilities.

Perceived productivity is a subjective criterion of what is thought to be productivity leading towards project completion. As time progresses and true productivity is reported, there will be changes in perceived and then estimated productivity. As these changes occur and a clearer picture becomes known, there will be a need to increase weighting on true productivity. It cannot be assumed that future productivity will be a reflection of past trends.

A significant influence on productivity can be as a direct result of managerial direction. Managers can reduce wasted time by exerting proper control (direction) over staff, ensuring that for example good time keeping and work practices are maintained—this is especially true during early stages of a project, where completion time is some time away and hence commitment low. It is easier to motivate people when they can see a completion goal.

Those working on critical path(s) activities are likely to be more appreciative of the urgency of their particular work. This is not so true of those working on other network paths, see PTGs discussed earlier.

6.4.3 PRODUCTIVITY—COMPUTER LISTING

```
Perceived_Producty_MF := Perceived_Producty_MF + T0 *
(Chg_Perceived_Prod_MF);
```

(Ref. File SD.PAS 544)

```
Team_in_Training_MF := Team_in_Training_MF + T0 * (-
Training_MF + Hiring_MF);
```

(Ref. File SD.PAS 548)

```
{ Table function: _Eff_Direct_Prodty = graph(Managerial
Direction) }
T[31,1]   := 0.000;  T[31,2]   := 1.000;  T[31,3]   := 0.100;
T[31,4]   := 0.500;  T[31,5]   := 0.655;  T[31,6]   := 0.755;
T[31,7]   := 0.823;  T[31,8]   := 0.877;  T[31,9]   := 0.910;
```

```
T[31,10] := 0.938;  T[31,11]  := 0.958;  T[31,12]  := 0.975;
T[31,13] := 0.993;  T[31,14]  := 1.000;
```

(Ref. File SDINT.PAS 443–448)

```
Perceived_Work_Rema_MF := Work_to_Complete_MF + Known_
Rework_MF;

Estimated_Product_MF := (Perceived_Producty_MF * Weighting
_on_Prod) + (1-Weighting_on_Prod) *
(Base_Productivity_MF);

Est_Work_Compl_MF := Project_Team_MF * Estimated_Product_
MF * Week_Hrs;

Forecast_Completion_MF := (TIME div 5) + (Perceived_Work_
Rema_MF / Est_Work_Compl_MF);

if ((Forecast_Completion_MF-Scheduled_Completio_MF) > (10
- XTOL)) then
Change_in_Schedule_MF := Schedule_Extension_MF / T0  else
Change_in_Schedule_MF := 0;

if ((Fract_Proj_Complete_MF>=(1-XTOL)) AND
(Completion_Date_MF <= (0+XTOL)) AND
(Completion_Date_MF>=(0-XTOL))) then
Chg_Complet_Date_MF :=  (TIME div 5) / T0  else
Chg_Complet_Date_MF := 0;

Ratio_Forcst_Sched_MF := Forecast_Completion_MF /
Scheduled_Completio_MF;
```

(Ref. File AUXRATES.PAS 133–147)

```
TableWithHiLoCapability(ManagerialDirection, _Eff_Direct_
Prodty, 31);
  TableWithHiLoCapability(Schedule_Pressure_MF, _Effect_
  of_SP_Prod_MF, 1);
  Productivity_MF := Base_Productivity_MF * _Effect_of_SP_
  Prod_MF * _Eff_Direct_Prodty;
  Chg_Perceived_Prod_MF := (Productivity_MF - Perceived_
  Producty_MF) * 0.2;
```

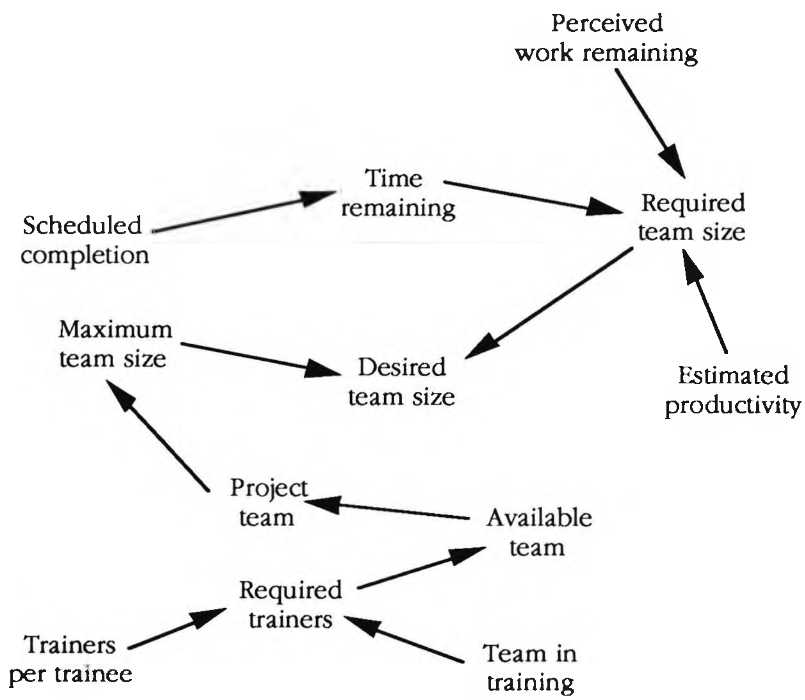
(Ref. File AUXRATES.PAS 151–154)

6.5 Project Team Size

PMs are likely to have a maximum team size allocated during the project planning stage, taking into account labour profile peaks computed initially by a PERT network. As LRs allocated to many projects are determined by average (total man–weeks work divided by CP length) labour consumption, LRA problems are made difficult to resolve.

PROMISS provides PMs with a facility to maintain a LR ceiling or produce a labour profile reflecting true LR requirements. From these figures there will be a derivative, desired team size, related to required team size, see Figure 6.4.

This in turn is determined by perceived work remaining. Two additional factors influencing required team size will be amount of time remaining and estimated productivity to complete outstanding work; time remaining being influenced by scheduled completion date and estimated productivity. PMs need to be aware of these factors, before assessing the total size of their project labour force.

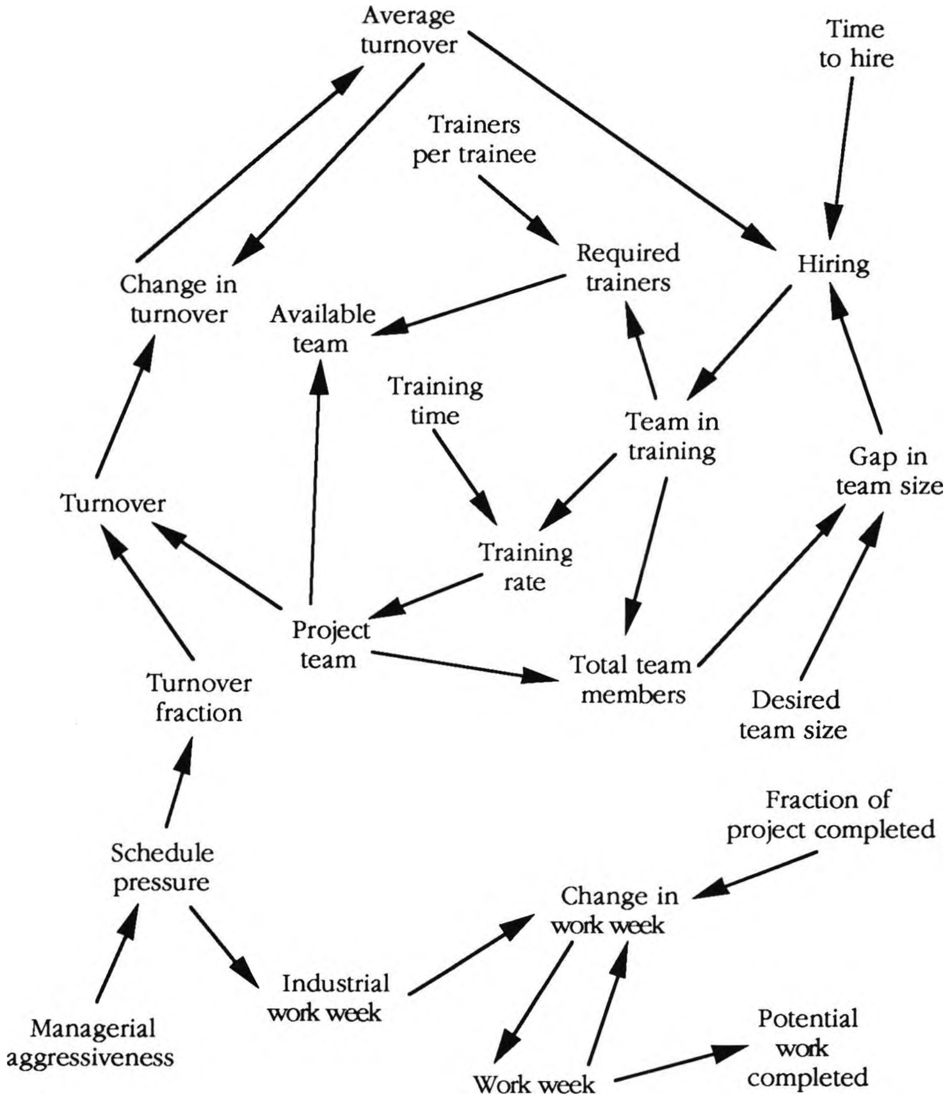


Simulation model concept, Part 4
Figure 6.4

Scheduled completion will determine time remaining to complete a project and hence, when compared with perceived work remaining, required team size. Maximum team size is a calculation produced from PERT—project team (LRs allocated against each activity) with an arbitrary 25% added to take into account extra work (work with faults), people not always work for the entire project duration and have less than 100% estimated productivity. Desired team size compares this figure with required team size to complete outstanding work. Whichever is least will be the desired team size.

Project team size, initially determined by PERT, will also need to take into account numbers of LRs available to produce productive work. PMs should know what the desired team size should be and any discrepancy between this and total team size will create a 'gap in team size'. This will affect the hiring rate (affected directly by average turnover and the time it takes to hire replacement team members). Before becoming part of a project team, there is a period required to ensure that new members are as familiar with the work as the existing team. These two LR components make up total team size.

Total project team size is made up of those working on the project and those in training (training costs are a budgeted cost set against the project, an example being specialist welders involved in work on a nuclear reactor). Once total and desired team size figures are compared, a positive or negative gap will be known. This determines increases or decreases in the rate at which new resources are hired. Negative changes in project team size will be a result of turnover rate, see Figure 6.5.



Simulation model concept, Part 5
Figure 6.5

The variable turnover fraction is related to schedule pressure; as it increases so does turnover rate. Schedule pressure is likely to increase intensity of work rate (productivity), a rate that may be unacceptable to some members of a project team—they leave! Such pressure can affect hours worked and hence quantity (and quality) of work completed.

As most LSPs are completed over a period of time, managers need to accept a turnover of project team members. Changes in productivity caused by this should be included in the model. If the turnover fraction increases, then a greater proportion of the project team will be in training, thereby reducing overall productivity. It should not be assumed that existing resources have one level of productivity and new members another. New members may have an unknown productivity which could significantly reduce (or increase!) that of existing team members. An example of this is a requirement for new resources to be "looked after" and an allowance made when calculating productivity, until they become familiar with the project work. The size of project team is therefore reduced by turnover rate and increased by training rate.

6.5.1 PROJECT TEAM SIZE—COMPUTER LISTING

```
Project_Team_MF := Project_Team_MF + T0 * (Training_MF -
Turnover_MF);
```

(Ref. File SD.PAS 546)

```
Team_in_Training_MF := Team_in_Training_MF + T0 * (-
Training_MF + Hiring_MF);
```

(Ref. File SD.PAS 548)

```
Work_To_Complete_MF := Work_To_Complete_MF + T0 * (-Work_
Completion_MF - Work_Comp_wFaults_MF + Return_of_ Work_
wFaults_MF);
Work_Week_MF := Work_Week_MF + T0 * (Chg_Work_Week_MF -
Puls_to_End_Project_MF);
```

(Ref. File SD.PAS 551–552)

```
{ Table function: _Ind_Work_Week_MF = graph(Schedule_
Pressure_MF) }
T[21,1]  := 0.000;  T[21,2]   := 2.000;  T[21,3]   := 0.200;
T[21,4]  := 40.00;  T[21,5]   := 40.00;  T[21,6]   := 40.00;
T[21,7]  := 40.00;  T[21,8]   := 40.00;  T[21,9]   := 40.00;
T[21,10] := 54.30;  T[21,11]  := 58.20;  T[21,12]  := 59.40;
T[21,13] := 59.70;  T[21,14]  := 60.00;
```

(Ref. File SDINT.PAS 373–378)

```
{ Table function: _Turnover_Fraction_MF = graph(Schedule_
Pressure_MF) }
T[26,1]  := 1.00000; T[26,2]  := 2.00000; T[26,3]  :=
                                          0.10000;
T[26,4]  := 0.00000; T[26,5]  := 0.00048; T[26,6]  :=
                                          0.00124;
T[26,7]  := 0.00212; T[26,8]  := 0.00292; T[26,9]  :=
                                          0.00342;
```

```
T[26,10] := 0.00368; T[26,11] := 0.00380; T[26,12] :=
                                                 0.00388;

T[26,13] := 0.00396; T[26,14] := 0.00400;
```

(Ref. File SDINT.PAS 408–413)

```
   begin
     Available_Team_MF := Project_Team_MF - Required_
     Trainers_MF;
     If CeilingOnResources then
       Available_Team_MF := InitAvailableTeam_MF;
   end;
```

(Ref. File AUXRATES.PAS 108–112)

```
Required_Trainers_MF := Team_in_Training_MF * Trainers_
per_Traine_MF;
```

(Ref. File AUXRATES.PAS 120)

```
Perceived_Work_Rema_MF := Work_to_Complete_MF + Known_
Rework_MF;
```

(Ref. File AUXRATES.PAS 133)

```
TableWithHiLoCapability(Schedule_Pressure_MF, _Turnover_
Fraction_MF, 26);
Turnover_MF := Project_Team_MF * _Turnover_Fraction_MF;
Chg_TO_MF := (Turnover_MF - Avg_Turnover_MF) / 12;

if (Fract_Proj_Complete_MF < (1+XTOL)) then
  begin
    TableWithHiLoCapability(Schedule_Pressure_MF, _Ind_
    Work_Week_MF, 21);
    Chg_Work_Week_MF := (_Ind_Work_Week_MF - Work_Week_MF)
    / 2
  end
    else Chg_Work_Week_MF := 0;

Maximum_Team_Size_MF := InProject_Team_MF * 1.25;

Time_Remaining_MF := MAX((Scheduled_Completio_MF - (TIME
div 5)), 2);

Required_Team_Size_MF := Perceived_Work_Rema_MF /
(Estimated_Product_MF * Time_Remaining_MF * Week_Hrs);

Desired_Team_Size_MF :=  MIN(Maximum_Team_Size_MF,
Required_Team_Size_MF);
```

(Ref. File AUXRATES.PAS 156–173)

```
Total_Team_Members_MF := Team_in_Training_MF +
Project_Team_MF;

Gap_in_Team_Size_MF := MAX((Desired_Team_Size_MF - Total_
Team_Members_MF), 0);

Hiring_MF := Avg_Turnover_MF + (Gap_in_Team_Size_MF /
Time_to_Hire_MF);
```

```
if (Fract_Proj_Complete_MF < (1+XTOL)) then
  Incr_CPW_MF := Total_Team_Members_MF  else  Incr_CPW_MF
  :=  0;
```

(Ref. File AUXRATES.PAS 179–186)

```
Potential_Work_Comp_MF := Available_Team_MF * Work_Week_MF
* Productivity_MF;
```

(Ref. File AUXRATES.PAS 190)

```
TableWithHiLoCapability(Schedule_Pressure_MF, _Fract_
Compl_Correct_MF, 11);
Work_Comp_wFaults_MF := Potential_Work_Comp_MF * (1 -
_Fract_Compl_Correct_MF);
```

(Ref. File AUXRATES.PAS 195–196)

```
Training_MF := Team_in_Training_MF / Training_Time_MF;
```

(Ref. File AUXRATES.PAS 214)

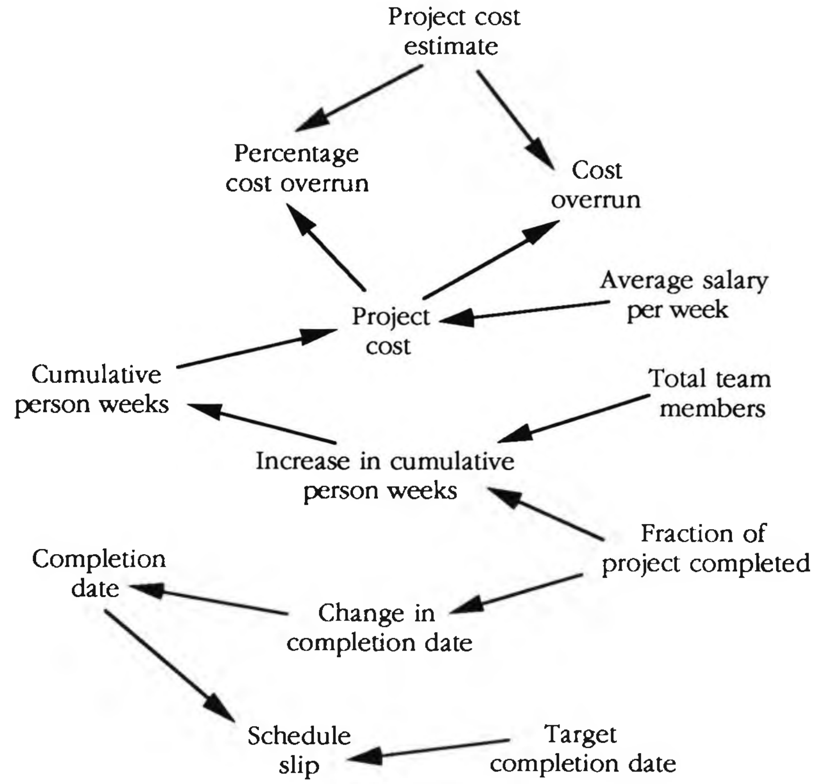
```
TableWithHiLoCapability(Schedule_Pressure_MF,
_Fract_Compl_Correct_MF, 11);
```

(Ref. File AUXRATES.PAS 216)

6.6 Project Cost

Most PERT network schedules provide a facility for cost estimation (for example Cost PERT). PERT, in PROMISS, does not do this as it was considered too time consuming for a development program. However, cost should have a significant influence on LRAs. Therefore, such a facility is included in the simulation model; it is at this stage of DSS usage that changes in LRs can be made and cost implications considered, see Figure 6.6.

Project cost (as a variable—assuming other physical costs remain unchanged irrespective of LRs used) is directly related to time (man weeks) and cost per week worked (either productive or non–productive i.e., where LRs are not fully utilised). The fraction of project completed will not only change the completion date but also determine cumulative man weeks worked. The original project cost estimate is determined by multiplying original man hours by cost per week (an initial rate of forty hours per week is used) for each resource type. Comparison between these two provides PMs with a cost overrun (which is either positive or negative). PMs can, when using PROMISS, determine cost changes as a result of their LRAs.



Simulation model concept, Part 6
Figure 6.6

6.6.1 PROJECT COST—COMPUTER LISTING

```
Project_Cost_Estim_MF := Project_Cost_Estim_MF;
```

(Ref. File SD.PAS 545)

```
InProject_Cost_Estim_MF := Target_Compl_Date_MF *
InProject_Team_MF * Avg_Salary_per_Week_MF;
```

(Ref. File SDINT.PAS 222)

```
Project_Cost_MF := Cumul_Person_Weeks_MF * Avg_Salary_
per_Week_MF;

Cost_Overrun_MF := Project_Cost_MF - Project_Cost_Estim
_MF;
```

(Ref. File AUXRATES.PAS 175–177)

```
PercentCost_Overrun_MF := ((Project_Cost_MF - Project_
Cost_Estim_MF) / Project_Cost_Estim_MF) * 100;
```

(Ref. File AUXRATES.PAS 200)

6.7 CONCLUSION

This chapter has described how significant elements make up the mathematical computer simulation model within PROMISS. Many other variables could be included, some included could be left out. Models are subjective and validation difficult, see Chapters 4 and 5, but these points should be considered in the light of the PROMISS concept—this is discussed further in Chapter 8.

CHAPTER 7

Discussion—Feedback and Feedforward Control

A Brigadier sent a message from the front line "Send reinforcements we are going to advance." Some time later, the message arrived at HQ, having been relayed several times, only to be responded to by the General "It appears they want us to send four and sixpence, they are going to a dance!"

Anonymous

7.1 INTRODUCTION

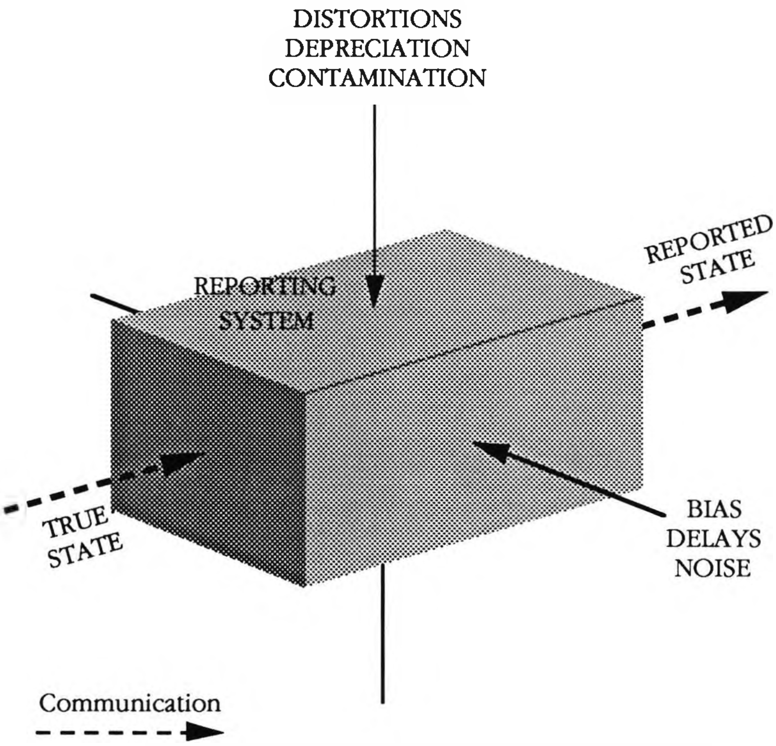
This chapter initially considers various aspects of control models used in a decision making environment. PMs in many situations find themselves having to make LRAs with less than perfect information. Although it is virtually impossible to have such perfect information, what is currently available could be improved, *before* a decision is made. Once this discrepancy has been accepted a Decision Support System (DSS) can take into account what is thought to be possible progress in the future, but may not be achieved because external influences were not taken into account.

PERT is essentially a feedback model, updated as work progresses, but unable to predict what *may* happen in the future. Therefore such an Information System (IS) i.e., PERT alone, could be usefully enhanced by taking into account possible interactions within a workforce environment. Control models and how they are used will be considered, leading to the use of PROMISS—an integration and use of both feedforward and feedback control.

Discussion continues in Chapter 8 with how PROMISS integrates data and turns it into decision making information for LRAs. Results from the PERT and simulation model (with various decision making scenarios) are also discussed.

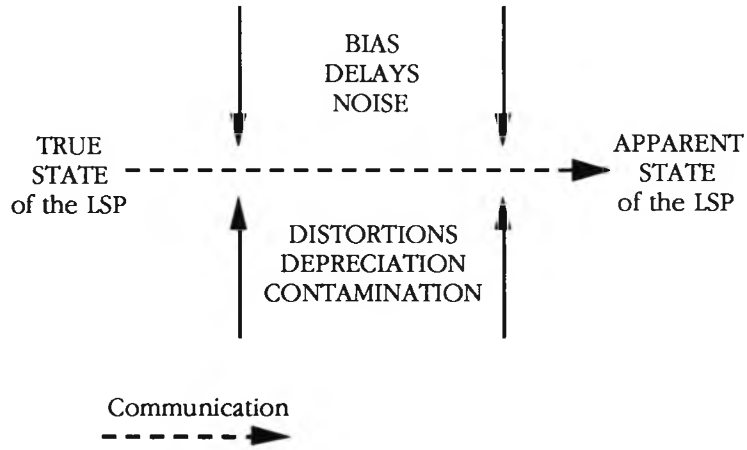
7.2 TRUE AND REPORTED STATES

An IS centres upon communication between what is a "true state" and the "reported state" (therefore such differences need to be taken into account when developing a model suitable for decision making). For an IS to be of any value, the central decision making model(s) should take into account the bias, delay, noise, distortion, depreciation and contamination which exist between these two states, see Figure 7.1. True state, in this situation, is the current reported state or a likely situation at some point in the future. In LSPs it is not unusual for the true state to be sometime in the past (from experience up to two weeks old—see Section 1.6 and Figure 1.8). A more useful reporting system would derive the true situation at the time that decisions regarding the future are made.



External influences between true and reported states
Figure 7.1

Discrepancies, through communication, are brought about by reporting links between what is, what is perceived to be and what should be. Many cases, from personal experience, would be sited where what is reported, bears little resemblance to the true state. An example is the marking down, incorrectly, of completed activities within the PERT, to enable other work to be started (see observation made on scheduling heuristics in Section 2.8.4). A reason for this is that many PERT software systems can produce a "plan–of–the–day" to guide LMs with work to be completed on a daily basis. Due to scheduling heuristics and available LRs, there is likely to be a difference between what *can* and what *should* be done. Figure 7.2 shows how differences between two LSP states can occur. This is similar to Figure 7.1 showing how the "true" state becomes an "apparent" state.

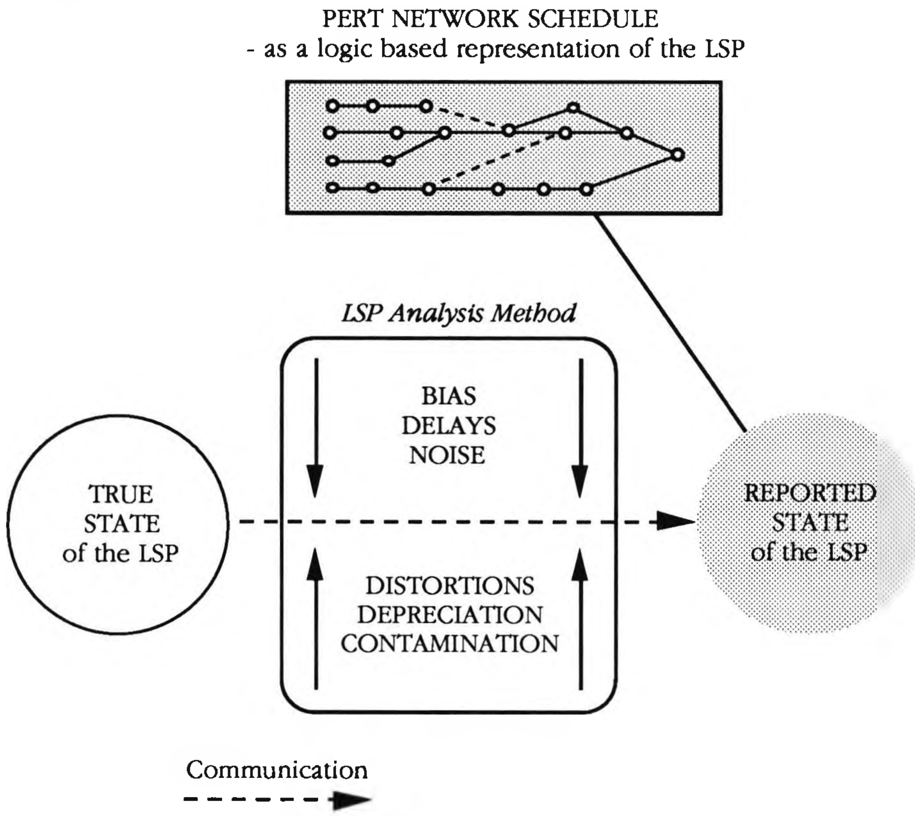


Causes of distortions between true and apparent states
Figure 7.2

This change of state, caused by interference, is inherent in any situation where there is a change in (or use of) medium between one state and another. Changes are caused by interference, some controllable by the analyst (for example, the method of analysis) and others due to the environment of the RW situation. With respect to this latter situation, a person views work being undertaken and reports the current state through a PERT network schedule. The reporting medium, in this example a computerised system, may not be able to accurately reflect the true state. An example of this is if an activity has been problematical and a number of tasks have had to be repeated. The amount of work completed as a percentage of activity completed may be difficult to report (reflect) in such a system. Such limitations, from which future courses of action will be determined, make decision making difficult (it is accepted that few decision making scenarios are made from a foundation of "perfect" information). In these circumstances, one needs to either reduce or take into account (because it is known) distortion and pollution between "true" and "apparent" states. If this were possible, through use of a model, then given an original state such a model would be able to present, a more reflective picture of the true situation.

7.3 Feedback—The Current Reporting System

Figure 7.3 adds to Figure 7.1 by introducing a PERT network schedule as a reflection of the "reported" state. The "true" state is shown with a solid outline representing a situation as it actually is, rather than "reported" as a grey symbol reflecting a non–positive (not so clearly defined) situation. See also Section 1.3 and Figure 1.3.

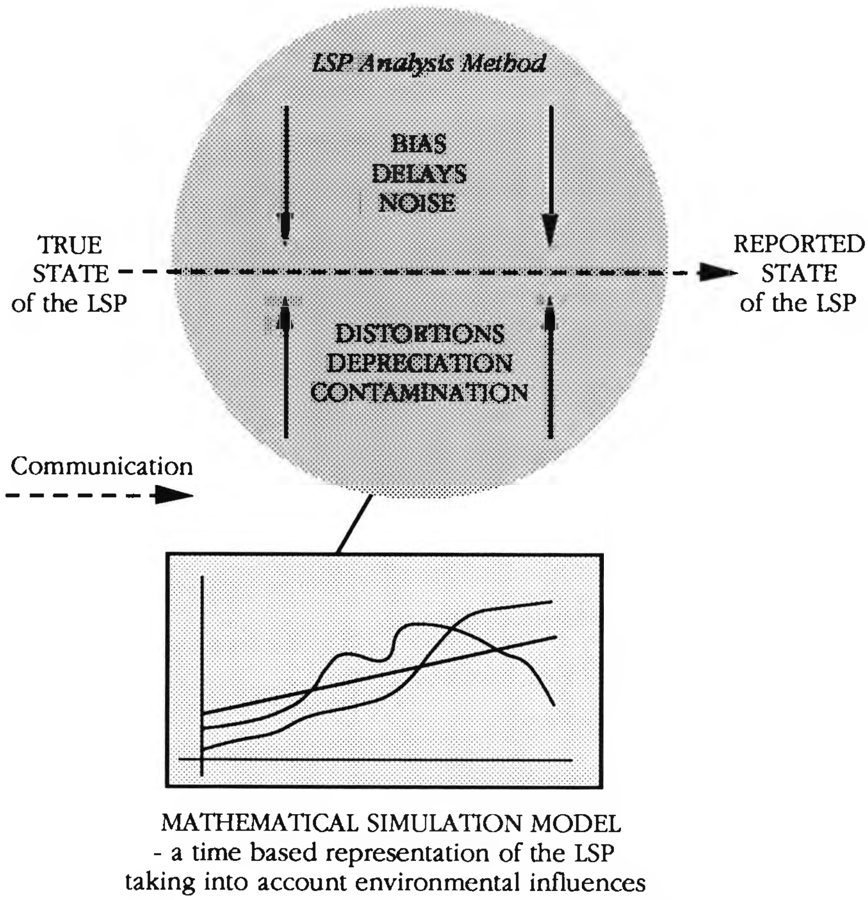


Reporting the true state
Figure 7.3

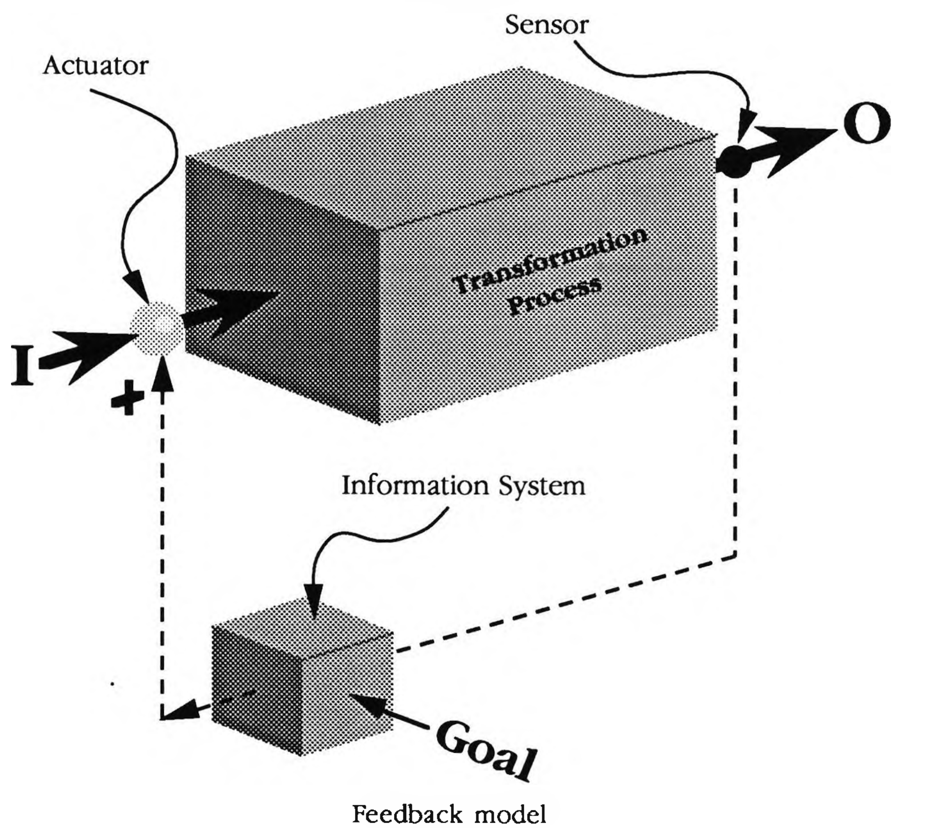
A LSP analysis method, as previously mentioned, includes factors that create distortion. To take this into account a means of reflecting such a state should be introduced in a reporting system so a PM can make decisions from a more genuinely "true" state. A mathematical simulation model is a model which can be programmed, taking into account interactions and influences causing a discrepancy between the two states. Such a model can reflect the analysis method, see Figure 7.4.

For some form of corrective control to take place, it is necessary to introduce a feedback path. A basic feedback model is shown in Figure 7.5. A transformation process changes inputs (I) into outputs (O). Inputs pass through an actuator controlling the amount going into a transformation process. A sensor monitors outputs and passes this information, as a signal, to an information system. Comparison (by a comparator) is made in the information system between a goal (or desired state) and actual state. If there is a difference, the actuator is adjusted to increase or decrease flow of inputs, enabling desired outputs to be produced. There are significant problems (with respect to managerial control) in such a control system:

- Control is a result of historical information;



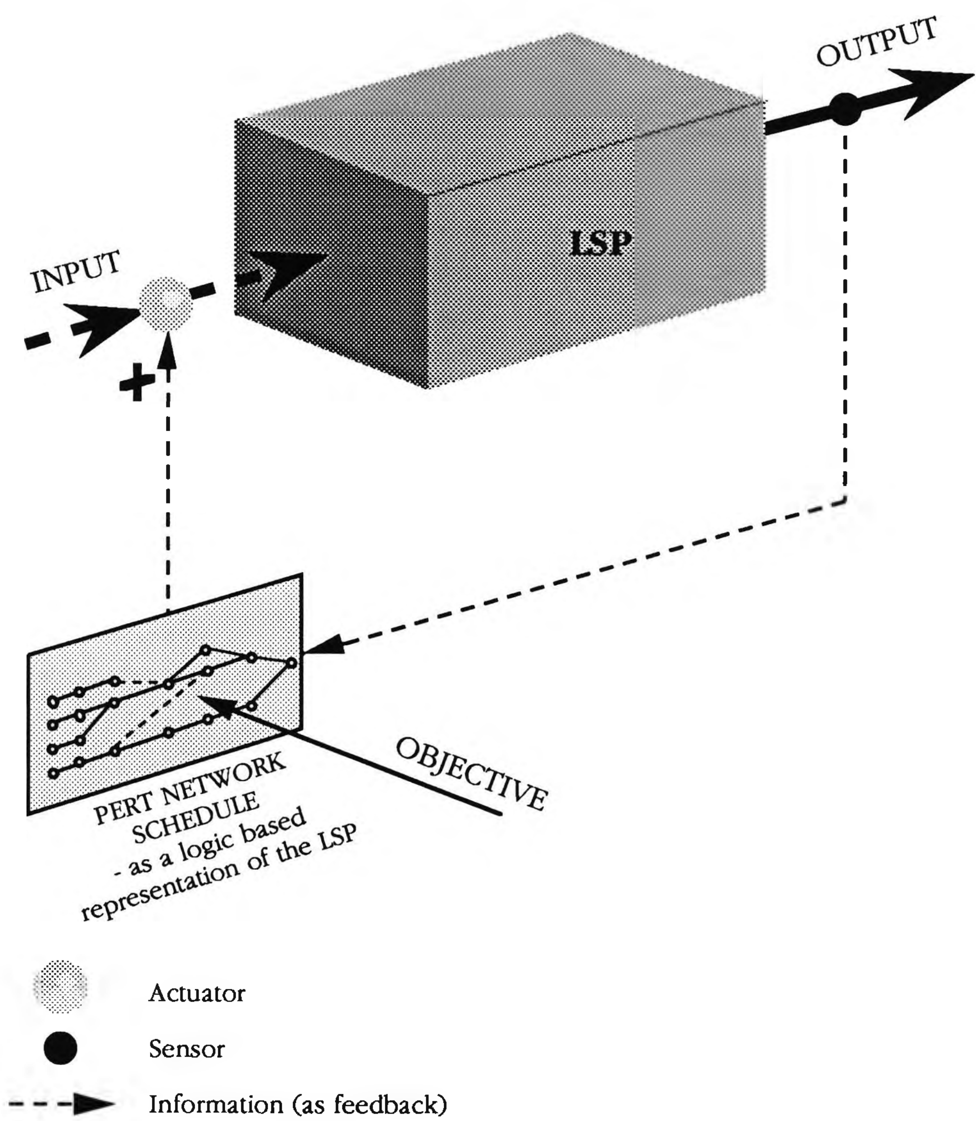
A means of including dilution between true and reported states
Figure 7.4



Feedback model
Figure 7.5

- control of the future is non–existent;
- response time of the feedback signal has a significant influence on how control and steady state outputs are maintained;
- in managerial situations there are many links in the feedback signal and hence distortions can (will) take place.

Currently, feedback reporting systems are reliant upon a PERT network schedule; this is a reflection of historical data/information (Flood and Stevens, 1987). Figure 7.6 shows a simple feedback diagram where inputs (resources) are transposed by the LSP into an output (in this case completion of the project). This output is monitored and compared to a PERT network which has an objective (the original plan of work). Should there be a discrepancy the

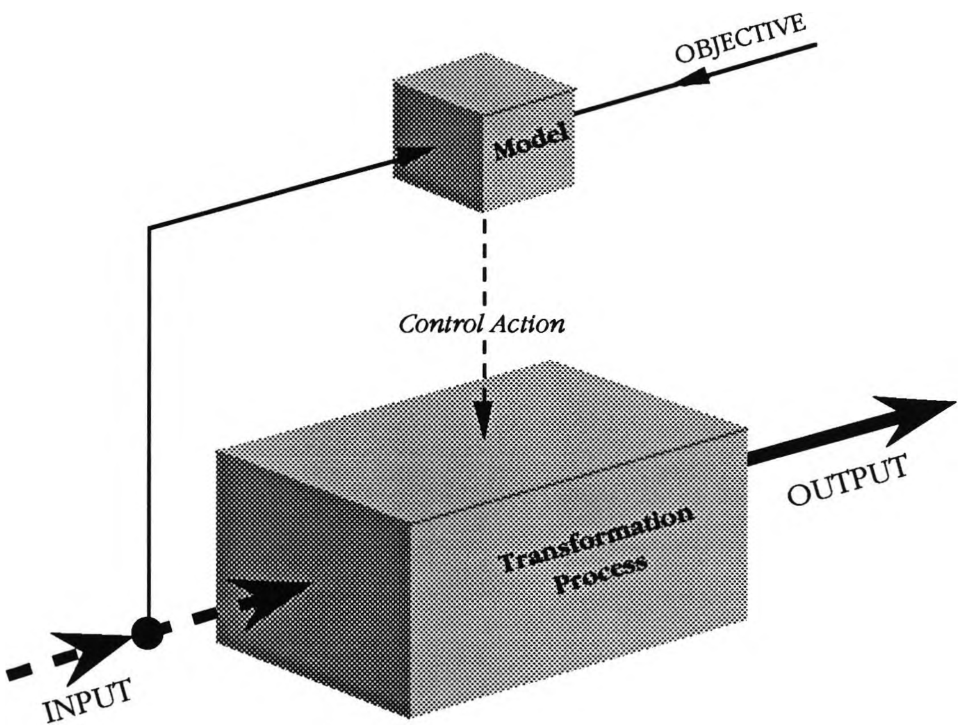


LSP report/control feedback
Figure 7.6

actuator is adjusted to increase or decrease inputs. Such a system, ignores the possible self correcting nature of elements within a LSP, for example, increase or decrease of productivity as a result of managerial pressure (an input not easily measured or even controllable). Only results (a change of output) can be measured. A better means of control is through the notion of feedforward control (Flood and Stevens, 1987; Stevens and Flood, 1988).

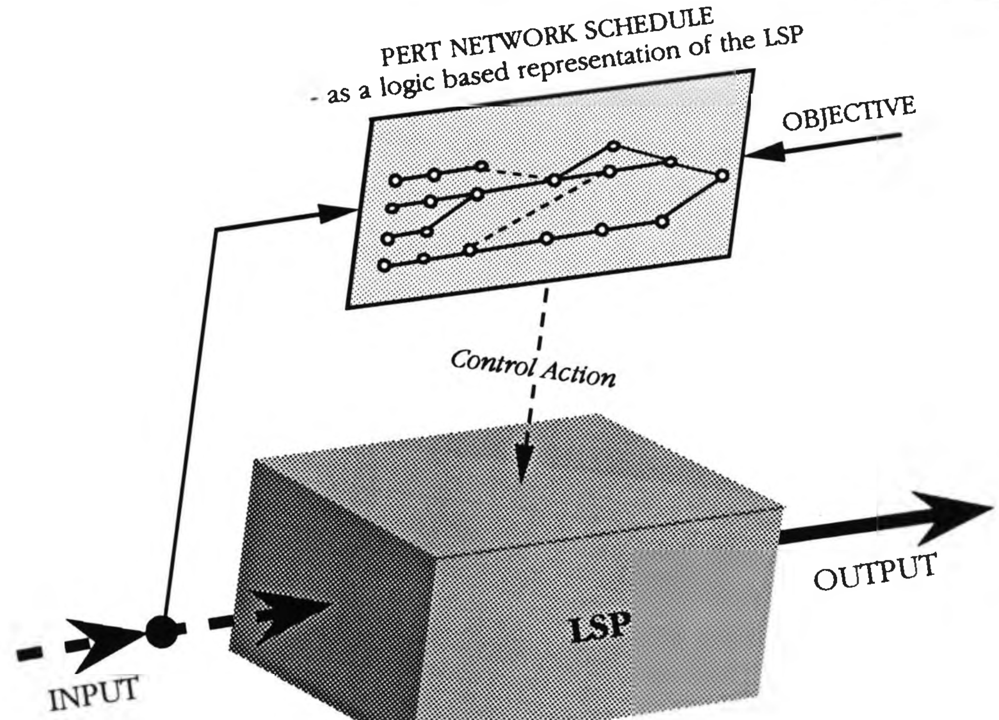
7.4 FEEDFORWARD—A NEW MEANS OF CONTROL?

Figure 7.7 shows inputs and outputs transformed by a LSP, as in feedback Figure 7.6, showing the notion of control through information derived from what has already been produced. PMs should employ a more effective means

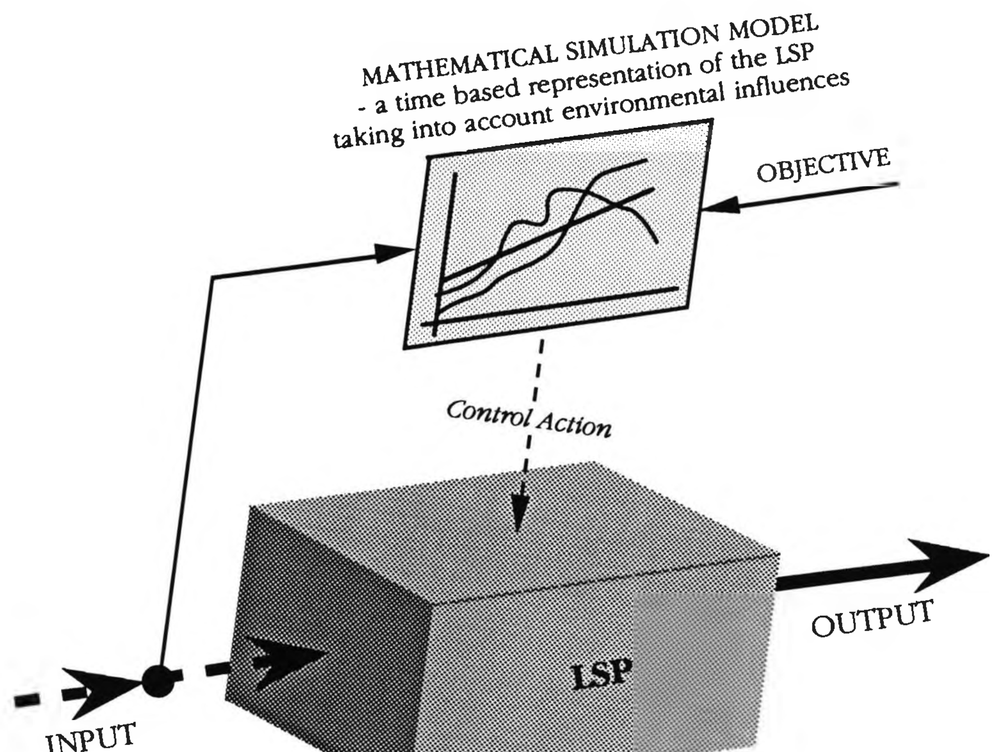


Simple feedforward control
Figure 7.7

of control action, using information about what is likely to happen, rather than what has already occurred. A means of deriving this control action on a LSP is by use of a model. Managers (PMs) or users can consider changes of input and try them with a model (a perceived situation(s) representation i.e., a LSP)



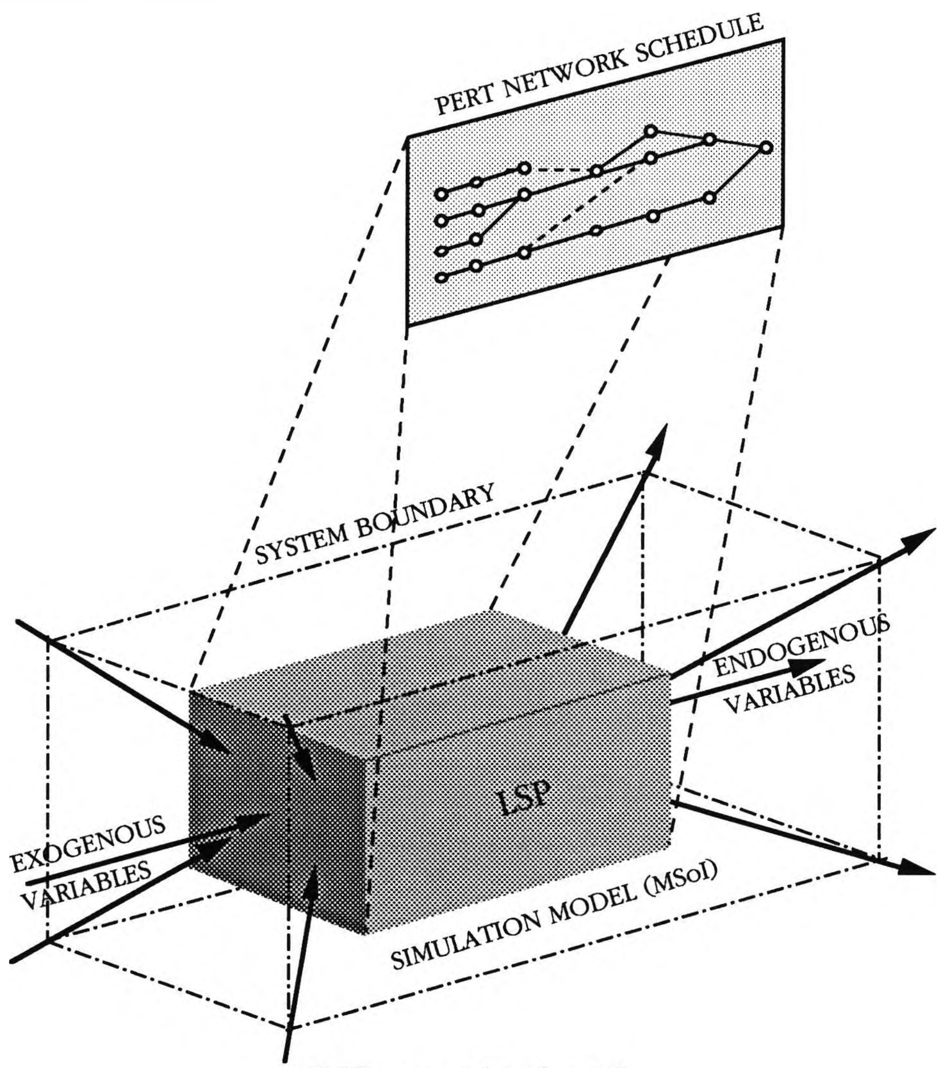
Feedforward control of a LSP using PERT
Figure 7.8



Feedforward control of a LSP using a mathematical simulation model
Figure 7.9

using "what if" scenarios to determine possible outcomes of such actions. Figures 7.7, 7.8 and 7.9 show this—control action being managerial action and not that of a model. This situation, see Figure 7.8, is to a certain extent theoretical, as the only model of a LSP is a PERT; limitations of this type of model are discussed in Flood and Stevens, 1987 and in Section 1.4.

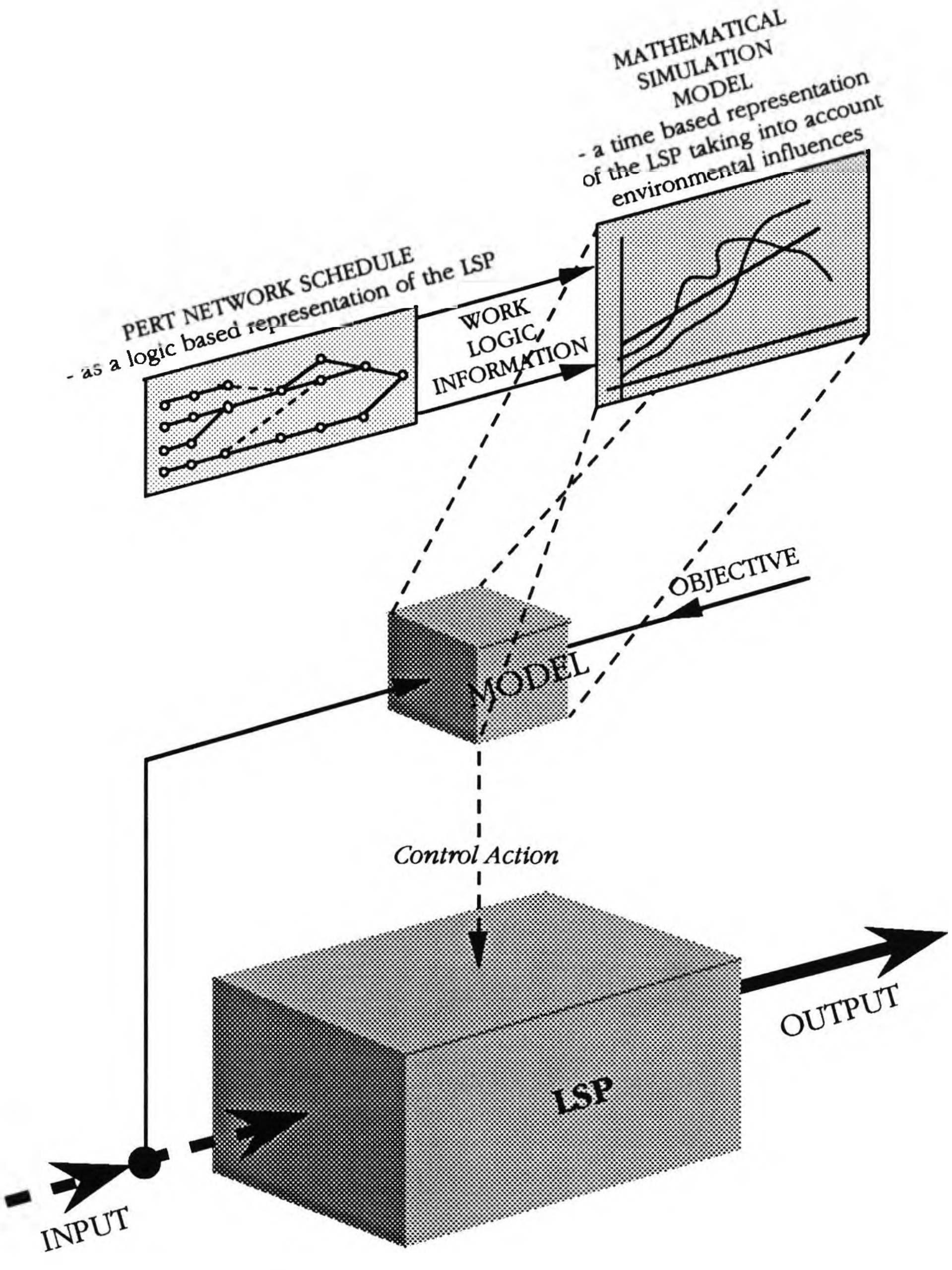
Feedforward control would be possible if a mathematical simulation model was used, as in Figure 7.9. If managerial control utilising feedforward is introduced, using a currently adopted PERT network schedule for a decision making model would be problematic, due to the inherent limitations mentioned earlier.



PERT as a model of an LSP
Figure 7.10

Should a PERT network schedule be used, then it should include exogenous and endogenous variables. A model of a situation (system) inside a boundary, if it is to be used in such a mode should include all the variety of a situation under control (Ashby's Law of Requisite Variety (LRV), see Section 4.6.5 and

7.6). PERT in its current state does not do so, see Figure 7.10 (same as Figure 4.1).



Feedforward control with PERT and mathematical simulation model
Figure 7.11

A PERT network schedule, as discused in Section 2.1, is an essential component in the control system of a LSP, but does not take into account the time based nature of work being undertaken. It also excludes internal and environmental influences that occur during work being undertaken (the PERT can only take part of these influences, through statistical means, in determining activity Actual Times derived from Optimistic; Pessimistic; and

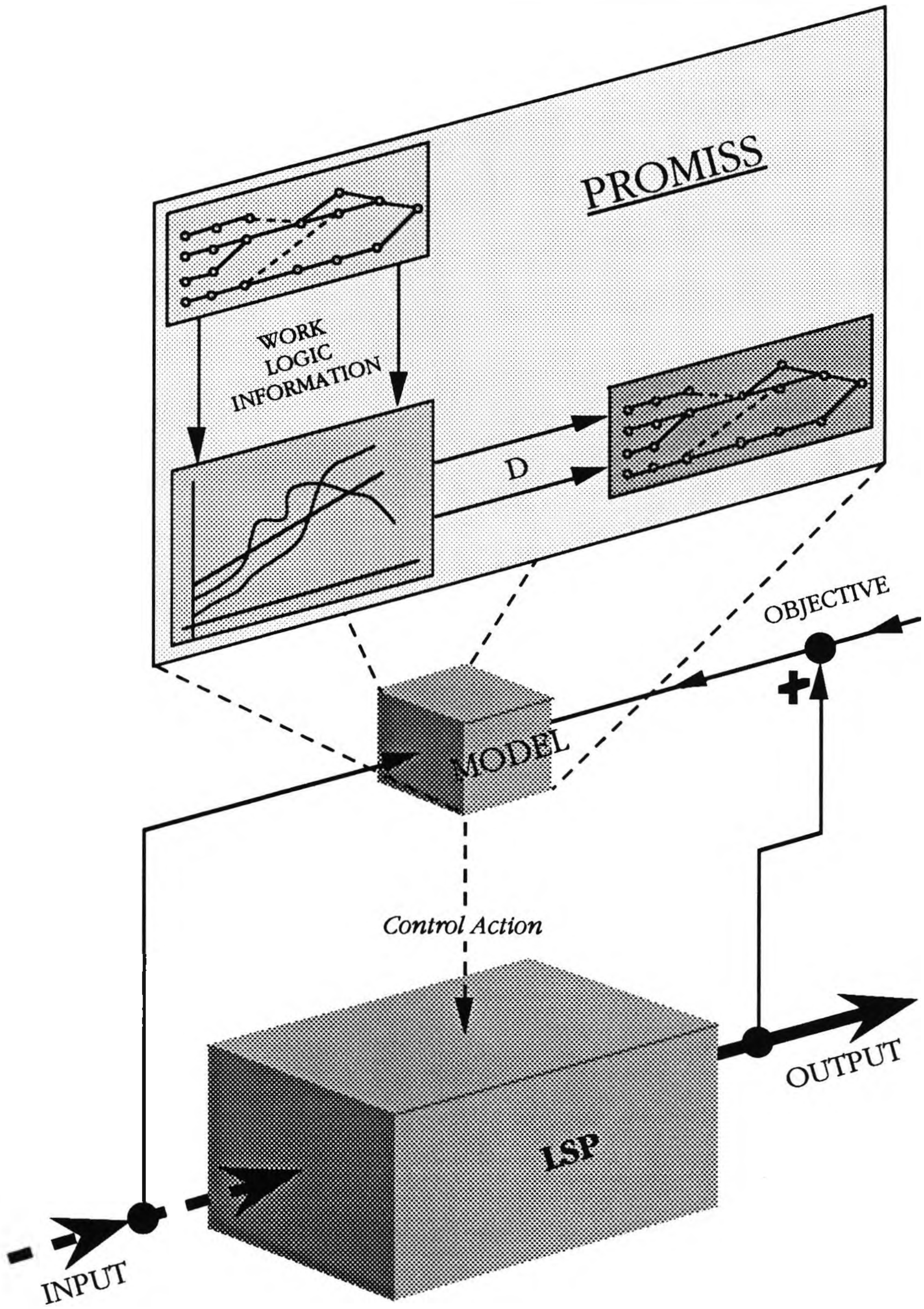
Time Most Likely), see Section 2.5.2. If the two user requirements of a LSP planning system are to be considered, see Section 7.6, then both a logic based plan and representation of all other influences have to be modelled. This situation, as shown in Figure 7.11, should provide sufficient information for reasonable feedforward control actions to be determined and implemented by PMs. This information will only provide an *indication* of what may happen, provided nothing (that has not been modelled) else changes.

Figure 7.11 shows how a PERT network schedule provides work logic information to a mathematical simulation model, taking into account other project environmental influences. Here external and internal influences of a project can be modelled, providing a more realistic picture to a decision maker. If this modelling method were used there would only be a one way flow of data, to be transformed into information by a simulation model. If PERT is not a true situation reflection, logically incorrect or not up to–date with what has happened, a GIGO situation will arise.

7.5 PROMISS—Feedback and Feedforward Control

Figure 7.12 is an extension of Figure 7.11 providing an outline of PROMISS—a better model of a LSP. PERT, as a base component, remains and maintains a central work plan, essential for those undertaking the work. This, as in Figure 7.11 feeds into a mathematical simulation model, work logic information, in the form of a data file. Once a number of "what if" scenarios have been run a new data file is created. This new file provides data to redraw a new logic work plan, a new PERT containing the same logic sequence, but providing realistic completion times for each activity/path under the changed conditions. This reflects what would/could occur when working in an LSP environment. PROMISS, therefore, considers the work logic sequence but does not accept that a CP, for example, will be a reflection of what is likely to happen in the future. Exogenous variables in the form of adding and removing LRs (at the direction of a PM) during the project duration can be taken into account. Endogenous variables are not considered, due to interactions of various elements inside the simulation model.

PMs need better guidance to enable them to be more fully aware of what should be and what is, but also what might be. Once these additional factors are acknowledged and modelled, and taken into account during the decision



Major Components of PROMISS

Original PERT Network Schedule

Revised PERT Network Schedule

Mathematical Simulation Model (MSM)

D Data link between MSM and PERT

Outline diagrammatic arrangement of PROMISS
Figure 7.12

making process, a more realistic future work plan can be developed. A number of "what if" scenarios can be undertaken with a model. Taking results of the most acceptable outcome, a user/PM can take this data and determine what the new logic plan is likely to be.

7.6 DISCUSSION

PMs need to have information that is more representative of the true situation they are controlling. They should be able to identify and know what is planned and exert feedforward control from a position of knowing what is likely to happen. Models used therefore need to provide this ability—the controller needs to have the same (or greater) variety as the controlled—Ashby's Law of Requisite Variety (Conant and Ashby, 1970).

PROMISS, as written for this research thesis, may not be the most efficient (it has been developed from scratch where subroutines (in the form of unit files) have had to be changed many times as a result of changes in algorithms). Future versions of PROMISS can be made to run more efficiently now that the basic concept has been shown to work.

Whilst PERT analysis can utilise A_t (as most project management software is able do) there are three other times that could be used; t_o, t_m and t_p. For a complete scenario of likely completion times of a project, an analysis would be needed for each activity time. A formula would be $S = \frac{-9}{2}[1-3^{n-1}]$ (worked out by Roudsari (personal communication, 1991); where S = number of scenarios and n = number of activities) and not that proposed by Vermeulen and Jongh (1976), see Section 5.9. Results of these iterations/scenarios could supply the best, average (or most likely) and worst cases. These would, however, hide from PMs the complexity of RW problems associated with running a project. Each scenario could be a reflection of what *may* happen, but none could be used for accurate forward thinking and anticipation of problems/situations. If all this information was available to a PM, it is likely to bury the best course of action (and associated decisions), rather than illuminating it. Various risk analysis techniques can be applied to such problems, but, the end result of such computer analysis would still be some form of uncertainty. This uncertainty would be even more dubious if weightings were applied to each activity completion time. Information from such iterations/scenarios would be sizably greater than standard (no-risk) analysis, even though it would be more comprehensive. Such comprehensiveness is due to considerations of time values between t_o and t_p in whatever time steps thought realistic/ feasible. Such iterations/scenarios, for reasons given above, would not be undertaken in project environments with currently used technology (due to the need for a large quantity of processing power, probably only currently available for military purposes).

A crucial success factor in implementing feedforward control is a realistic reflection of the model used. Computer simulation models now have influence on how situations are managed—this is likely to increase rather than diminish in the future.

Simulation modelling is (and probably always will be) very subjective. It is difficult to predict, by computer, what *may* happen in a future situation. Even in areas where there are almost perfect facilities for providing forecast scenarios, it is likely to be just a guide to what could happen, provided nothing else changes. For example, consider peoples' fixation with the weather. The United Kingdom weather forecasting system has almost perfect information (they have many years of historical records), satellites, ground weather stations, Atlantic Ocean weather ships, rain detecting radar posts and reports from both ships and aircraft travelling around the Northern Hemisphere. All this data and information is analysed with complex mathematical models, by versions of the world's most powerful computers (Cray) and they still cannot guarantee what the weather will be like in twelve hours time! What hope is there for managers in complex industrial situations, to be able to guarantee that results from computer simulation models will be reliable enough to eliminate a human decision maker? What should be considered is that any model, bearing some resemblance to a situation, from which various scenarios can be undertaken, is better than nothing at all. The test for a model is whether it helps.

Traditional managers, in the 1990s, can be forgiven for such short-sightedness, but unfortunately the most experienced managers are those who have reached their last day at work, before retiring (they have gained maximum experience). Today's managers are expensive to educate/train and then obtain the very necessary experience needed to perform tasks professionally. Management science has gone someway to enhance and shorten the maturing stages of a manager. In conjunction with this has been an increasing reliance on information technology—the ability to manage information. Large scale projects and data (information) overload appear to go hand–in–hand. Network scheduling programs and powerful computers are able to provide managers with every conceivable format of information presentation and in great quantities. Unfortunately, many managers find themselves overwhelmed by data, at times when there is a need for specific help in making decisions (information). Situations can arise when more data is available there is less likelihood it will be used.

7.7 CONCLUSION

This chapter has discussed some of the issues associated with feedback and feedforward control. Both these topics can be explored in greater depth, but it was only necessary to discuss points relevant to development of PROMISS and integration of both types of control.

Feedback control is essentially control of the past—historical information as a basis of controlling future courses of action has limited (but in the case of LSPs, essential) use. Feedforward control is control of the future—an essential issue for efficient management. Why do PMs (as do many managers) still control from the past? One reason is there are few means of doing otherwise. Use of feedforward models for control in management (especially management of LRAs) is still in its infancy.

This chapter has discussed how PROMISS was conceived. Use of feedforward control has many advantages, but relies upon a model to help provide a controller with the same (or greater) variety as the controlled—LRV. Such models are subjective, see Chapters 4, 5 and 6. However, some form of feedforward model must be better than reliance for control on a feedback model. PROMISS integrates both feedback and feedforward as a means of decision support for PMs. LMs are able to use a currently available logic guide to help them understand sequences of events to complete a project or task. PMs are able as forward thinking managers, to predetermine what may happen if certain courses of action were to be implemented. Chapter 8 continues the discussion of how PROMISS was developed and how it can be used in management of LSPs.

CHAPTER 8

Integration—PROMISS

8.1 INTRODUCTION

This chapter looks at how PROMISS becomes an integrated DSS, for use in LSPs. PROMISS was conceived as an enhancement to currently used project software—this was the research hypothesis.

PERT, as discussed in Chapters 1 and 2, provides LMs and PMs with a logic guide of how to accomplish a task/project. Many (if not most) LSPs appear to overrun in both time and cost. A significant reason for this is the unique nature of most LSPs—they are generally "one–offs" and/or difficult to make comparisons with other work. As a result of this situation, managers (both LMs and PMs) have difficulty in knowing what has been done, where they are and where to go. Planning, before commencement of a project, provides a guide, but for most situations will be wrong at Start Day + 1 (if one resource does not materialise, changes to plans need to be made) and as work progresses other unexpected (unforeseen) changes ensure plans are no longer plans, but guides.

This DSS makes use of feedback ***and*** feedforward control as a means of enhancing information for control. PROMISS utilises these means of control by integrating PERT (feedback) and a computer simulation model (feedforward).

8.2 PERT VERSUS SIMULATION

PERT is a logic based model made up of a number of activities arranged in order, in a combination of serial and parallel formats, see Chapter 2. PROMISS ensures that this essential plan/guide is available and can be used by those requiring it—LMs, managers of LRs and people carrying out physical tasks of completing a project.

PERT has been a modelling technique used in project management environments since the late 1950s, see Chapter 1. Its efficiency as an aid to controlling project resources is limited to: a) logical sequence of activities being correctly identified and positioned; b) checking past work against that plan; and c) identifying essential path(s)/sequence of packages of work needing constant allocation of resources (activities on CPs).

A mathematical computer simulation model is a time based modelling technique able to provide dynamic future information (derived from differential equations). Provided a model has been conceived as a reflection of a situation and nothing else changes (exogenous influences), ***reasonable*** predictions can be made.

Computer simulation models have been developed into a "near art form" since the last world war, when computers became readily accessible to commercial/industrial organisations. Although use of such models has had a significant influence on many aspects of industrial life, they have not been widely accepted in the area of project management. Reliance on PERT is almost total. PROMISS, by integrating a computer simulation model with a PERT attempts to enhance the attributes of both.

Enhancement is brought about by ensuring PERT remains fundamentally unchanged but introduces another model to provide a more realistic picture of what ***may*** happen in the future. PERT, as discussed in Chapter 2 and 7, is poor at providing feedforward control information. Dynamic computer models, however, can provide such control information (it provides a guide to what should happen, ***not*** what is likely to happen). A significant enhancement is in the area of LRAs—an expensive and significant influence on the success of LSP completion. Currently, with PERT the only means of planning optimal utilisation of LRs is by resource levelling.

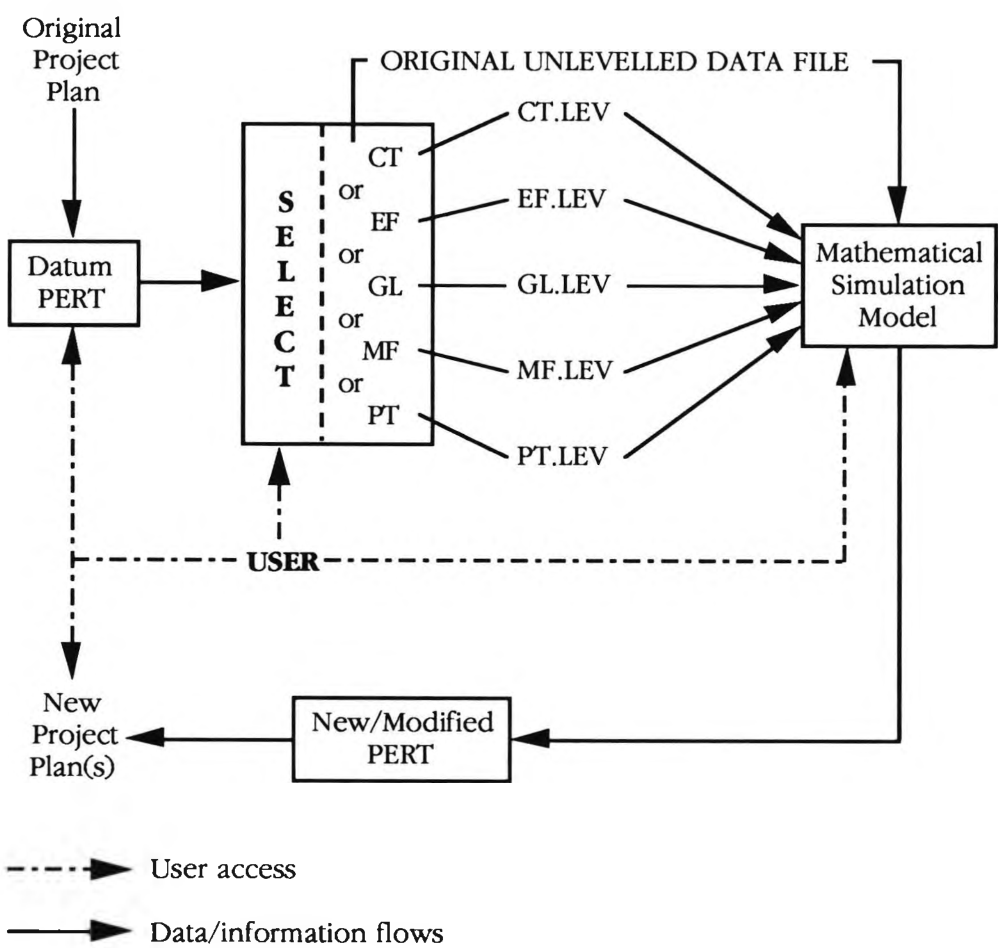
Most PERT programs are able to provide a resource levelling facility (see Chapter 4) but for LSPs this can be time consuming (in both managers' and computer time). If it is possible to try out a number of LRA scenarios by a quicker means, before trying it out on a PERT, time will be served and their is a greater likelihood that readily available information will be used. A computer simulation is able to provide such a facility and PROMISS the integration.

8.3 INTEGRATION

PROMISS can be used for trying out a number of possible options, choosing one that "best fits" a situation and then produce a new logic plan (PERT) for those carrying out project tasks. For this facility to be provided, there initially needs to be two computer models: one containing a logic plan and one to dynamically project into the future. This is how PROMISS was developed—see Chapters 4 and 6.

The basic work plan/structure originates by drawing out a logical sequence of events to complete a task. This, as is current practice, makes use of a PERT network schedule. PERT inside PROMISS, performs this task. A user can derive, after analysis, both early and late start and finish times as well as identifying CPs. Once this has ben achieved a resource levelling exercise should be undertaken—this "optimises" LRs by making use of time tolerance available on non–critical paths (NCP) i.e., float. Section 8.4 discusses resource levelling times.

Figure 8.1, an extension of Figure 3.1, shows how PROMISS works and integrates PERT with a mathematical simulation model. A user (PM) provides data/information to a PERT and then decides to either level a prioritised resource or leave it unlevelled. Once this data has been computed it can then



PROMISS—old to new PERT
Figure 8.1

be passed into a mathematical simulation model. Here users can interact and change a number of variables, either as initial values (Screen 63 Appendix 5) or interactive variables (Screen 64 Appendix 5). A number of scenarios can then be run to find a suitable outcome from possible LRAs. After these scenarios have been simulated a new revised (modified) PERT can be produced. This provides both PMs and LMs with a plan showing:

- planned and actual start and finish;
- planned and actual activity lengths.

Running a number of scenarios (maximum of ten per session) is quicker than changing original PERT data and levelling each option. Section 8.4 below describes times to resource level with an accepted "near optimal" algorithm.

8.4 Resource Levelling Times

Appendix 6 provides detailed results of levelling exercises when Mechanical Fitters are given priority and data provided by network schedule PERT1B (datum network). As this was undertaken on a third generation PC (an 80386 based CPU rather than a first generation 8088/6—see Appendix 3 for various CPU specifications) of average performance (a clock speed of 20Mhz but without a numeric co-processor)—a comparison can be made with results making use of an 8086 based computer. Such a comparison demonstrates that a choice of computer *may* influence use of a PERT for decision making. The less time it takes to provide results, the greater its likely use. Both 8086 and 80386 PCs are common in industrial companies and therefore should be readily available for running of PROMISS. An 80386 based PC is, for this research, a benchmark PC. Table 8.1 below provides a synopsis of levelling times for all five resources utilised in the datum project.

Original Start Times are derived from datum network PERT1B (generating *x*.LEV data files) and changes given as a percentage improvement in resource allocation (utilisation). These data files provide a user with details of activities improved by shifting start times and offering recommended start times. These tables are reproduced in Appendix 7 for each resource priority.

Appendix 6 also provides a comparison of the same levelling exercise, carried out with priority to Mechanical Fitters, on a comprehensive range of PCs with differing clock speeds (some with a numerical coprocessor) and CPUs. It is

Table 8.1
Synopsis of levelling times for PERT1B running on an 80386 based PC without a numerical coprocessor

Priority Resource Type	File Name	Time to Level	Improvement
Constructive Trades	1BC.LEV	12 h 43 min	8.13%
Electrical Fitters	1BE.LEV	14 h 42 min	10.78%
General Labour	1BG.LEV	15 h 35 min	8.10%
Mechanical Fitters	1BM.LEV	14 h 7 min	9.35%
Painters	1BP.LEV	11 h 40 min	8.48%

not necessary to provide times for all resources as any percentage differences between them will be pro-rata. Precise terminal time for an 8088 based PC is an extrapolation, as length of processing would exclude use of this computer as a viable option for use with PROMISS.

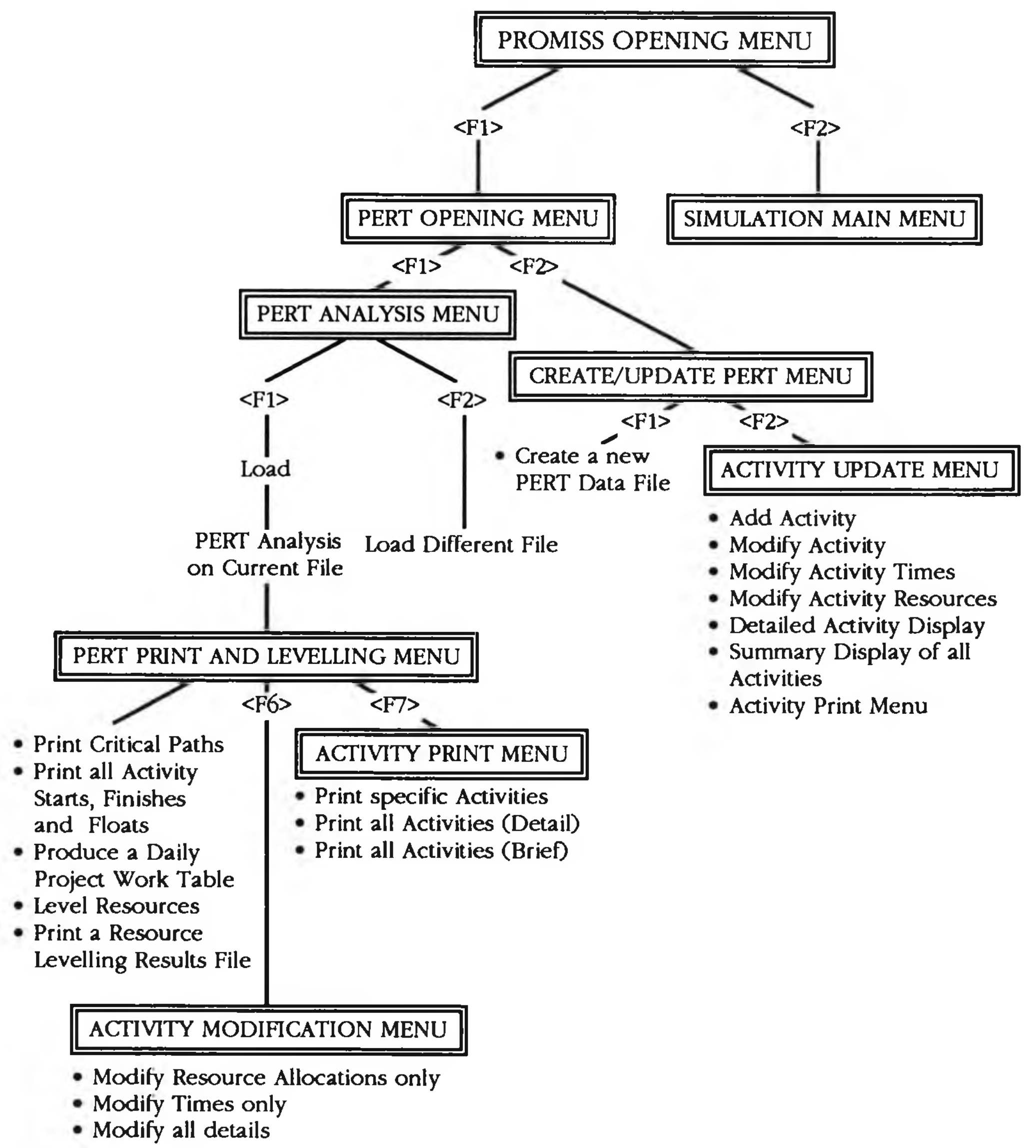
Use of PROMISS on a computer capable of providing sufficient information, within a reasonable time–frame, is likely to be used more frequently than most currently used PERT software. Most of this software can only provide changes in work logic, rather than assimilated expectations of various operational changes, routinely carried out during a project. PROMISS enables PMs to run various options, selecting the best (not necessarily optimal), an then reschedule work logic plans for those undertaking future tasks. Speed in obtaining results from both PERT network schedule and simulation components of PROMISS is likely to be crucial in the decision making process.

8.5 Using PROMISS

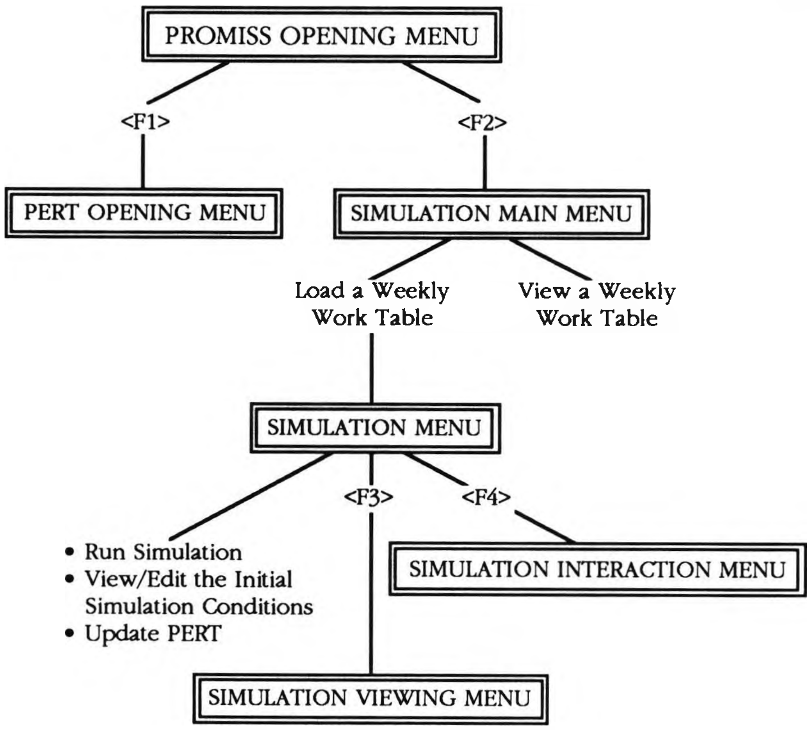
PROMISS is a development system and does not contain enhancements such as GUI and WIMP found in commercial packages. Likewise data/information outputs are numerical tables and not graphics. These can be added later should a need arise. PROMISS is menu/function key driven and users are able to follow the menu tree structures in Figures 8.2 (PERT) and 8.3 (Simulation).

PROMISS is started by typing "promiss" (it is not case sensitive) as this boots a central or root program connecting PERT and simulation models together. Users have an option of running either model from the Opening Menu (see Appendix 5: PROMISS Users Guide). The simulation model can only run if there is a PERT data file available, otherwise PERT must be run first. Figure 8.4 provides brief details of the first twenty five activities from PERT1B, a datum test PERT of one hundred and ninety nine activities (two hundred being the limiting number imposed on PROMISS users). Figure 8.5 shows full details of the first six PERT1B activities.

Once a PERT has been created there is sufficient data for various scenarios to be run using the simulation option. When a user has decided on a scenario of LRAs a new PERT can be generated by running the feedback option from SIMULATION MENU (<F5>).



PROMISS PERT menu tree structure
Figure 8.2



PROMISS simulation menu tree structure
Figure 8.3

Act No	Act Name	Preceding activities		Expected time (H)	Variance
1	1–2			43.57	4.13
2	3–4			52.18	8.51
3	5–6			64.45	2.05
4	7–8			106.13	3.36
5	9–10			181.29	11.90
6	11–12			40.00	0.00
7	13–14			110.36	9.30
8	15–16			81.50	0.00
9	17–18			128.97	52.56
10	2–19	1		174.08	6.08
11	4–20	2		118.87	2.72
12	6–20	3		70.30	7.47
13	8–10	4		42.58	1.25
14	10–21	5	13	94.80	0.00
15	12–22	6		131.98	3.48
16	14–23	7		108.50	0.00
17	16–24	8		131.98	1.28
18	18–25	9		53.62	1.32
19	24–26	17		12.48	1.52
20	D27–26	37		0.00	0.00
21	22–28	15		6.70	0.00
22	28–29	21		23.67	0.90
23	21–29	14		104.50	0.00
24	8–30	4		42.60	0.00
25	30–31	24		63.82	2.15

Activity information (brief)—PERT1B
Figure 8.4

Activity Number 1 Name 1–2
Length (Hours) 43.57 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation	1	1	0	0	2
Expected (Hours)	43.57	23.70	0.00	0.00	84.53
Variance	0.49	0.40	0.00	0.00	0.49
Critical Resource	Cr				

Activity Number 2 Name 3–4
Length (Hours) 52.18 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation	1	0	2	0	1
Expected (Hours)	52.18	0.00	104.33	0.00	39.27
Variance	0.00	0.00	1.21	0.00	1.21
Critical Resource	Cr				

Activity Number 3 Name 5–6
Length (Hours) 64.45 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation	1	2	0	0	0
Expected (Hours)	64.45	72.85	0.00	0.00	0.00
Variance	1.03	0.17	0.00	0.00	0.00
Critical Resource	Cr				

Activity Number 4 Name 3–4
Length (Hours) 106.13 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation	1	3	0	0	2
Expected (Hours)	23.82	203.33	0.00	0.00	212.25
Variance	0.30	0.81	0.00	0.00	0.15
Critical Resource					Cr

Activity Number 5 Name 9–10
Length (Hours) 181.29 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation	0	0	2	1	0
Expected (Hours)	0.00	0.00	362.58	83.27	0.00
Variance	0.00	0.00	6.50	0.81	0.00
Critical Resource			Cr		

Activity Number 6 Name 11–12
Length (Hours) 40.00 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation	1	0	0	0	0
Expected (Hours)	40.00	0.00	0.00	0.00	0.00
Variance	0.00	0.00	0.00	0.00	0.00
Critical Resource	Cr				

Activity information (detailed)—PERT1B
Figure 8.5

8.6 Feedback to a New PERT

Once PMs have chosen a suitable scenario of LRAs he will need to provide (and know himself) what the new network will look like. PROMISS provides this facility from the SIMULATION MENU. Basis of this computation is with the use of Newton's divisional/interpolation routine provided with Borland's Turbo Pascal Numerical Toolbox—it had to be heavily modified to provide results for all PERT scenarios. Memory management has however restricted maximum length of a simulated network to five hundred days (approximately 20% longer than PERT1B).

This interpolation routine views simulation results and relates them back to the original PERT. The simulated length of a project is made ***equivalent*** to its CP(s) and the two labour profiles compared. This computation provides the actual start and finish for each activity together with new lengths, see Table 8.2 for results of the first twenty activities.

Table 8.2
Sample of PERT feedback results (PERT1B)

Feedback File: PERT1B.FBK

Act No	Activity Name	Length (D) Planned	Actual	Planned Start	Finish	Actual Start	Finish	
1	1-2	43.57	58.82	0.00	43.57	0.00	58.82	
2	3-4	52.18	71.07	0.00	52.18	0.00	71.07	
3	5-6	64.45	86.51	0.00	64.45	0.00	86.51	
4	7-8	106.13	153.97	0.00	106.13	0.00	153.97	
5	9-10	181.29	272.90	0.00	181.29	0.00	272.90	
6	11-12	40.00	54.05	0.00	40.00	0.00	54.05	
7	13-14	110.36	163.86	0.00	110.36	0.00	163.86	
8	15-16	81.50	111.97	0.00	81.50	0.00	111.97	
9	17-18	128.97	207.19	0.00	128.97	0.00	207.19	Cr
10	2-19	174.08	253.85	43.57	217.64	58.82	312.67	
11	4-20	118.87	191.61	52.18	171.06	71.07	262.68	
12	6-20	70.30	132.64	64.45	134.75	86.51	219.15	
13	8-10	42.58	84.67	106.13	148.71	153.97	238.64	
14	10-21	94.80	114.32	181.29	276.09	272.90	387.22	
15	12-22	131.98	209.54	40.00	171.98	54.05	263.59	
16	14-23	108.50	150.63	110.36	218.86	163.86	314.49	
17	16-24	131.98	194.65	81.50	213.48	111.97	306.61	
18	18-25	53.62	67.05	128.97	182.58	207.19	274.24	Cr
19	24-26	12.48	18.37	213.48	225.97	306.61	324.98	
20	D27-26	0.00	0.00	238.87	238.87	343.29	343.29	

8.7 ADVANTAGES OF PROMISS

PROMISS provides an enhancement to currently available PERT software used in projects of any size. This enhancement is a mathematical based computer simulation model. Once an initial project plan is loaded into the PERT a PM is able to analyse, as he does currently, when each activity could be started and completed. LRs were added with each activity. This information is estimated when each activity work package was broken down by trade estimators.

However, planned utilisation of these resources is unlikely to be followed, for example when insufficient resources are provided, see Section 2.8.4. Work will be completed as and when resources become available (after completion of other activities) but in an order dictated by the PERT logic plan. The actual future consumption of resources is however difficult to predict.

One means of analysing these requirements, against availability is by resource levelling—this option is provided by PROMISS. Resource levelling will provide an "optimal" labour profile in ideal conditions, i.e., conditions that will enable those activities to be completed as per original schedule. Unfortunately the reality of working in a LSP environment is generally different. This makes it very difficult to determine where a project is (with respect to likely completion of outstanding work) and if planned completion dates will be achieved. Environmental influences that affected the past may not be present in the future; if they are present they may be more or less influential over the remaining project duration.

A means of analysing this situation is by making use of a simulation model. Once PMs have either "optimised" a labour profile by levelling or maintained an unlevelled schedule (keeping planned start and finish dates at their earliest possible times) he can find out how such influences affect project completion and consumption of resources. PROMISS accepts this original data and subjects it to influences described in Chapter 6. PMs may, as a result of this new information, change LRAs by interacting through a number of interactive scenario simulations. From these results an acceptable plan can be provided. Such a plan (labour profile) does not however provide sufficient information for LMs to complete a project. They will need a logic plan showing when activities should be started and finished and in what sequence.

A new logic plan is provided by PROMISS through use of a feedback option available once simulation data files are selected by a user. Information is similar to Table 8.2. Managers (both project and line) are able to compare planned (levelled, including those unaffected by this procedure or unlevelled as per original plan prior to simulation) against a likely "actual" plan. Once managers are made aware of likely deficiencies/shortfalls or additional costs or savings as a results of LRAs corrective action can be taken—feedforward control.

Such a DSS can now be used to enhance the quality of decisions needed to ensure a project completes to original criteria, for example utilisation of resources and/or cost. PROMISS integrates and provides from the original project data two model outputs to help in this decision making process—such a facility is not available on any other currently used project computerised information system.

8.8 CONCLUSION

This chapter has shown how integration of two types of computer simulation models: logic and time based, can enhance information availability for both PMs and LMs. As a research development system PROMISS is large and complex, but it does provide sufficient information, as an enhancement on what is currently available, to PMs reliant on only a PERT network schedule. PROMISS provides a considerable amount of information and data and it is not possible to reproduce it all in here—it can be easily produced from the menu driven program(a significant advantage of an information support system is to provide information as and when required).

Sufficient details have been given here to provide an overall picture of information availability and what it could be used for. Such a provision confirms that the research hypothesis has been proven.

Chapter 9

Conclusion

The author, at the beginning of this research, stated that from his experience of working in project management environments there was a shortage of suitable information for use in labour resource allocations. A research hypothesis to develop an enhancement to Program Evaluation and Review Technique network schedules should be explored. This, after three and a half years work, has been accomplished.

The first two chapters of this thesis described how large scale projects were managed, organised and used network schedules. Reluctance of software houses to release their program source codes necessitated a network schedule being written, see Chapter 3. This version does not contain the many "user friendly" and reporting interfaces found in professional packages. It does however serve a purpose of processing data into a format suitable for planning a project of up to two hundred activities.

Chapters 4 and 5 discussed computer modelling and System Dynamics as a methodology for developing a mathematical computer simulation model, as an enhancement to a network schedule. Consideration was given to both positive and negative aspects of such modelling techniques. System Dynamics has been used as a systems methodology for a number of years, but its popularity has waned as other methodologies gained prominence. Use of such a methodology does not detract from the value of this research but could maintain its usefulness for situation understanding, leading to problem solving, (Stevens, 1990; 1991). Chapter 5 provided a clearer understanding of SD because consideration was given to original philosophical issues related to this methodology. Notice was taken of Albert Einstein's quote at the beginning of this thesis. As time progresses, many foundational issues become diluted and polluted as conceivers lose prominence and other people make their own interpretations. Chapter 5 readdressed this situation and provided a means of constructive advancement. As there was a goal to be achieved (that of completing a large scale project, within a criterion of optimising available labour resources) a Hard Systems Methodology, utilising computer simulations, was considered appropriate—this has been proven by the success of this research.

Chapter 6 briefly described how a hybrid model of a project was developed. Many variables introduced and interlinked may be considered subjective and likely to be disputable—the author supports this argument. In defence of this model, it should be agreed that ***all*** models are subjective, unless they are an ***exact*** replication of reality. PROMISS (PROject Managers Information Support System) was developed by considering a number of key variables that ***may*** influence successful completion of a project. This was undertaken by looking at physical and information flows inside a project environment.

PROMISS was conceived as a development project to enhance decision making for project managers. There is an essential requirement of network schedules remaining at the core of any large scale project information system—it provides an important logic guide for those undertaking tasks leading to completion. Such a system was developed over thirty years ago and little has changed, even though many different modelling techniques have been developed for other spheres of work. One such development is mathematical simulations running on personal computers. Such models can enhance the quality of decision making, but currently many are flawed for reasons discussed in Chapter 4.

Integration of logic and time based models centres upon the concept of feedback and feedforward information for control. This was discussed in Chapter 7. One means of introducing feedforward control is through the use of a model, in this case a computer simulation. Given the limitations of such a technique, enhancing one form of modelling with another, to provide a clearer picture of what is *likely* to happen, could lead to improved decision making by project managers.

One of the most difficult aspects of managing large scale projects is the efficient direction of labour resources. PROMISS was developed to help this situation by providing a means of trying out a number of scenarios *before* implementing them. By completing all scenarios available with PROMISS, decision makers can be provided with a clearer understanding of: 1) where they are now; 2) why they may not be where they should be; and, 3) how a future situation may evolve. PROMISS, like any decision making model can provide subjective reasons of what may have happened, but more importantly for PMs, as managers of the future, what may happen if certain courses of action are taken. Chapter 8 describes how this decision support system works.

Finally there is a requirement to determine areas of further development. After the first twelve months of this research a publication was produced (Stevens and Flood, 1988) stating that the ultimate development goal for PROMISS would be as an Expert System. This still remains. PROMISS, as a concept, works as a logical extension for currently available decision support systems. Future development should be in the provision of providing project managers with information and help in diagnosing reasons for deviations in plans. Likewise, there should be a means for providing results of past decisions through use of an Expert System database. These points, together with a more "user friendly interface" and better means of constructing a simulation model should be areas for enhancement and development. This simulation modelling aspect should follow *Stella* for the Apple Macintosh (or with the use of Windows for the PC) format and allow users to introduce their own variables. PROMISS, as a prototype decision support system for use by large scale project managers, provides a firm foundation for this advancement.

This concludes the thesis on development of an alternative decision making information system for labour resource allocations: Integration of logic and time based large scale project computer simulations—PROMISS.

References and Bibliography

If I had read as much as other men, I should know no more than they.

Thomas Hobbes,
reported by Aubrey, *Brief Lives.*

ACKOFF, R. L., (1962), *Scientific Method: Optimizing Applied Research Decisions*, Wiley: New York.

ACKOFF, R. L., (1967), Management Misinformation Systems, *Management Science*, December, 14(4), pp. B–147–56.

ACKOFF, R. L., and SASIENI, M. W., (1968), *Fundamentals of Operations Research*, Wiley: New York.

ACTON, F. S., (1970), *Numerical Methods That Work*, Harper and Row: New York.

ADELSBERGER, H. H., (1986), Introduction to artificial intelligence, *Proceedings of the Conference on Intelligent Simulation Environments–Simulation Series*, 23–25 January, 17(1), pp. 16–22, San Diego, California.

AHUJA, H. N., and NANDAKUMAR, V., (1985), Simulation Model to Forecast Project Completion Time, *Journal of Construction Engineering and Management*, December, 111(4), pp. 325–42.

ALFELD, L. E., and GRAHAM, A. K., (1976), *Introduction to Urban Dynamics*, The MIT Press: Cambridge, Massachusetts.

ALLISON, G. T., (1971), *Essence of Decision*, Little, Brown and Company: Boston.

ANDRETTA, M., LUGARESI, M., MAZZANTI, P., and SERRA, R., (1986), MAPS: An Expert Advisor For The Qualitative Analysis of Dynamic Systems, In: Aracil, J., Machuca, J. A. D., and Karsky, M., (eds), *System Dynamics on the Move, Proceedings of the 1986 International Conference of the System Dynamics Society*, pp. 1049–60, Seville, Spain.

ANKLESARIA, K. P., and DRESNER, Z., (1986), A Multivariate Approach to Estimating the Completion Time for PERT Networks, *Journal of Operational Research Society*, August, 37(8), pp. 811–5.

ANSOFF, H. I., and SLEVIN, D. P., (1968), An Appreciation of Industrial Dynamics, *Management Science*, March, 14(7), pp. 383–97.

AONUMA, T., (1964), On a partition algorithm for critical path method, *Journal of the Operations Research Society of Japan*, 7(2), December, pp. 49–67.

BADIRU, A. B., (1988), Towards the Standardization of Performance Measures for Project Scheduling Heuristics, *IEEE Transactions on Engineering Management*, 35, May, pp. 82–9.

BALCI. O., and NANCE, R. E., (1985), Formulated Problem Verification as an Explicit Requirement of Model Credibility, *Simulation*, August, 45(2), pp. 76–8.

BALKE, W. M., HAMMOND, K. R., and MEYER, G. D., (1973), An Alternative Approach to Labour Management Relations, *Administrative Science Quarterly*, 18, pp. 311–27.

BALLARD, D. H., and BROWN, C. M., (1982), *Computer Vision*, Prentice–Hall: Englewood Cliffs, New Jersey.

BANKS, J., CAPTAIN, D. G., and CAPTAIN, S. P. S., (1988), Modeling process, validation, and verification of complex simulations: A survey, *Proceedings of the Conference on Methodology and Validation–Simulation Series*, 6–9 April, 19(1), pp. 13–8, January 1988, Orlando, Florida.

BATTERSBY, A., (1970), *Network Analysis for Planning and Scheduling*, (3rd Ed.), Macmillan.

BELLMAN, R. E., and DREYFUS, S. E., (1962), *Applied Dynamic Programming*, Princetown University Press: Princetown, New Jersey.

BENNINGTON, G. E., (1974), Applying network analysis, *Industrial Engineering*, January, 6(1), pp. 17–25.

BERG, C. C., (1974), Individual Decisions Concerning the Allocation of Resources for Projects with Uncertain Consequences, *Management Science*, September, 21(1), pp. 98–105.

BENKU, T., (1986), The Runge–Kutta Methods, *Byte*, April, pp. 191–210.

BILDSON, R. A., and GILLESPIE, J. R., (1962), Critical path planning—PERT integration, *Operations Research*, 10(6), November–December, pp. 909–12.

BIHAN, P., (1987), *Turbo Prolog: An Introduction to Artificial Intelligence*, Wiley: Chichester.

BILES, W. E., (1988), Design of computer simulation experiments, *Proceedings of the Conference on Methodology and Validation—Simulation Series*, 6–9 April, 19(1), pp. 7–12, January 1988, Orlando, Florida.

BITTIE, L. R., (1962), Pocket PERT, *Factory*, 120(7), July, p. 81.

BLAIR, R. J., (1963), Critical path resources simulation and scheduling, *The Institution of Electrical and Electronic Engineers Transactions on Engineering Management*, 10(3), September, pp. 100–3.

BLALOCK, H. M., (1971), *Causal Models in the Social Sciences*, Macmillan: New York.

BOULANGER, D. G., (1961), Program evaluation and review technique (PERT)—a case study application with analysis, *Advanced Management*, 26(7), July–August, pp. 7–12.

BOWLES, K., (1978), *Microcomputer Problem Solving Using Pascal*, Springer: Berlin.

BRACHMAN, R. J., (1979), On The Epistemological Status of Semantic Networks, In: Findler, N. V., (ed.), *Associative Networks: Representation and Use of Knowledge by Computers*, Academic Press, pp. 3–50.

BRENNER, M. E., (1963), Selective Sampling—a Technique for Reducing Sample Size in Simulation of Decision Making Problems, *Journal of Industrial Engineering*, 14, pp. 291–6.

BREITENECKER, F., (1983), The Concepts of Supermacros in Today's and Future Simulation Languages, *Mathematics and Computers in Simulation*, 25, pp. 279–89.

BREWER, G. D., (1973), *Politicians, Bureaucrats, and the Consultant*, Basic Books: New York

BRITISH STANDARDS INSTITUTION, BS 6046, (1984), *Use of network techniques in project management*:
Part 1, *Guide to the use of management, planning, review and reporting procedures*;
Part 2, (1981), *Guide to the use of graphical and estimating techniques*;
Part 3, *Guide to the use of computers*;
Part 4, *Guide to resource analysis and cost control*.

BRITISH STANDARDS INSTITUTION, BS 4335, (1987), *Terms used in project network techniques*.

BRITTING, K. R., (1976), Backward Integration of System Dynamics Models—A Useful Validation Test?, *TIMS Studies in the Management Sciences*, 2, pp. 141–9.

BRUNER, J. S., GOODNOW, J. J., and AUSTIN, G. A., (1956), *A Study of Thinking*, Wiley: New York.

BUFFA, E. S., and DYER, J. S., (1978), *Essentials of Management Science/Operations Research*, Wiley: New York.

BURDEKIND, R., and MARSHALL, S. A., (1972), The Use of Forrester's System Dynamics Approach in Urban Modelling, *Environmental Planning*, 4, pp. 471–85.

BURDEN, R. L., FAIRES, J. D., and REYNOLDS, A. C., (1981), *Numerical Analysis*, Prindle, Weber and Schmidt: Boston, Massachusetts.

BURGER, R., and FISCHER, M., (1986), An Expert System for Project Management, *Proceedings of The International GPM/Internet Symposium: Project Management Software Application–Implementation–Trends*, 15–18 June, pp. 209–33, Garmisch, FRG.

BURGESS, A. R., and KILLEBREW, J. B., (1962), Variation in Activity Level on a Cyclical Arrow Diagram, *Journal of Industrial Engineering*, March–April, pp. 76–83.

BURGHES, D. N., and WOOD, A. D., (1980), *Mathematical Models in the Social, Management and Life Sciences*, Wiley: Chichester.

BURNS, J. R., (1977), Converting Signed Digraphs to Forrester Schematics and Converting Forrester Schematics to Differential Equations, *IEEE Transactions on Systems, Man, and Cybernetics*, October, SMC–7(10), pp. 695–707.

BURNS, J. R., and MALONE, D. W., (1974), Optimization Techniques Applied to the Forrester Model of the World, *IEEE Transactions on Systems, Man and Cybernetics*, March, SMC–4(2), pp. 164–71.

BURT, J. M., Jr., and GARMAN, M. B., (1971), Conditional Monte Carlo: A Simulation Technique for Stochastic Network Analysis, *Management Science*, November, 18(3), pp. 207–17.

CARLSON, E. D., (1979), An Approach for Designing Decision Support Systems, *Data Base*, Winter, pp. 3–15.

CARLSON, S., (1951), *Executive Behaviour*, Stromberg: Stockholm.

CHAN, K. H., (1984), Decision Support Systems for Human Resource management, *Journal of Systems Management*, April, pp. 17–25.

CHARNES, A., and COOPER, W. W., (1962), A network interpretation and a directed sub–dual algorithm for critical path scheduling, *Journal of Industrial Engineering*, 13(4), July–August, pp. 213–19.

CHARNIAK, E., and McDERMOTT, D., (1985), *Introduction to Artificial Intelligence*, Addison–Wesley: Reading, Massachusetts.

CHESTNUT, H., (1965), *Systems Engineering Tools*, Wiley: New York.

CHILD, J., (1984), *Organization: A Guide to Problems and Practice*, Harper and Row.

CHORFAS, D., (1962), *Systems and Simulation*, Academic Press: New York.

CHURCHMAN, C. W., ACKOFF, R. L., and ARNOFF, E. L., (1957), *Introduction to Operations Research*, Wiley: New York.

CLARK, C. E., (1961a), The Greatest of a Finite Set of Random Variables, *Operations Research*, 9, pp. 145–62.

CLARK, C. E., (1961b), The optimum allocation of resources among the activities of a network, *Journal of Industrial Engineering*, 12(1), January–February, pp. 11–7.

CLARK, C. E., (1962), The PERT Model for the Distribution of an Activity Time, *Operations Research*, May–June, 10(3), pp. 405–6.

CLEMENTATION, A. T., (1963), *Extended Control & Simulation Language*, c/e. Cam.Ltd: Birmingham.

CLINGEN, C. T., (1964), A modification of Fulkerson's PERT algorithm, *Operations Research*, 12(4), July–August, pp. 629–32.

CODDINGTON, E. A., and LEVINSON, N., (1955), *Theory of Ordinary Differential Equations*, McGraw–Hill: New York.

COHEN, M. D., MARCH, J. G., and OLSEN, J. P., (1972), A garbage can model of organizational choice, *Administrative Science Quarterly*, 17, pp. 1–25.

CONANT, R. C., and ASHBY, W. R., (1970), Every good regulator of a system must be a model of that system, *International Journal of System Science*, 1(2), pp. 89–97.

COOK, T. M., and JENNINGS, R. H., (1979), Estimating a Project's Completion Time Distribution Using Intelligent Simulation Methods, *Journal of Operational Research Society*, December, 30(12), pp. 1103–8.

COOPER, L. L., and COOPER, M. W., (1981), *Introduction to Dynamic Programming*, Pergamon Press: Elmsford, New York.

COYLE, R. G., (1977), *Management System Dynamics*, Wiley: Chichester.

COYLE, R. G., (1983), The technical elements of the system dynamics approach, *European Journal of Operational Research*, 14, pp. 359–70.

COYLE, R. G., (1985), The use of optimization methods for policy design in a system dynamics model, *System Dynamics Review*, 1, pp. 88–91.

COYLE, R. G., (1986), Comment on "System Dynamics and Operational Research: An appraisal." *European Journal of Operational Research*, 23, pp. 403–6.

CRISSEY, B. L., (1974), A Conceptualized Framework for Selecting Quality and Level of Aggregation for Dynamic Computer Simulation Runs, *Proceedings of the 1974 Summer Simulation Conference*, 9–11 July, Houston, Texas.

CROOKES, J. G., (1982), Simulation in 1981, *European Journal of Operation Research*, 9, pp. 1–7.

CROOKES, J. G., BALMER, D. W., CHEW, T. S., and PAUL, R. J., (1986), A Three–Phase Simulation System Written in Pascal, *Journal of Operational Research Society*, June, 37(6), pp. 603–18.

CURRENT, J., (1988), The Minimum–Covering/Shortest Path Problem, *Decision Sciences*, 19, Summer, pp. 490–503.

CUYPERS, J. G. M., and RADEMAKER, O., (1974), An Analysis of Forrester's World Dynamics Model, *Automatica*, 10, pp. 195–201.

DAELLENBACH, H. G., GEORGE, J. A., and McNICKLE, D. C., (1983), *Introduction to Operations Research Techniques*, (2nd Ed.), Allyn and Bacon Inc.: Newton, Massachusetts.

DANIEL, J. W., and MOORE, R. E., (1970), *Computation and Theory in Ordinary Differential Equations*, Freeman.

DAVIS, E. L., (1986), Fifth generation simulation, *Proceedings of the Conference on Intelligent Simulation Environments—Simulation Series*, 23–25 January, 17(1), pp. 113–7, San Diego, California.

DAVIS, E. W., (1966), Resource allocation in project networks—a survey, *Journal of Industrial Engineering*, 17(4), April, pp. 177–88.

DAVIS, E. W., (1968), *An Exact Algorithm for the Multiple Constrained–Resource Project Scheduling Problem*, PhD Thesis, Department of Administrative Sciences, Yale University.

DAVIS, E. W., (1974), Networks: resource allocation, *Industrial Engineering*, April, 6(4), pp. 22–32.

DAVIS, E. W., and HEIDORN, G. E., (1971), An Algorithm for Optimal Project Scheduling Under Multiple Resource Constraints, *Management Science*, August, 17(12), pp. B–803–16.

DAVIS, E. W., and PATTERSON, J. H., (1975), A Comparison of Heuristic and Optimum Solutions in Resource–Constrained Project Scheduling, *Management Science*, April, 21(8), pp. 944–55.

DAVIS, G. B., and OLSEN, M. H., (1984), *Management Information Systems: Conceptual Foundations, Structure and Development*, (2nd Ed.), McGraw–Hill: Singapore.

DAVIS, K. R., and McKEOWN, (1984), *Quantitative Models for Management*, (2nd Ed.), Kent Publishing Company: Boston, Massachusetts.

DAVIS, L., and LAWRENCE, P. R., (eds.), (1977), *Matrix*, Addison–Wesley: Reading, Massachusetts.

DAVIS, R., (1982), Meta–Rules: Reasoning about Control, *Artificial Intelligence*, 15, pp. 179–222.

DAVIS, R., and LENAT, D.B., (1982), *Knowledge–Based Systems in Artificial Intelligence*, McGraw–Hill: New York.

DAVIS, W., and McCORMACK, A., (1979), *The Information Age*, Addison–Wesley: Reading, Massachusetts.

DEAN, B. V., (ed.), (1985), *Project Management: Methods and Studies*, North Holland: New York.

DECKRO, R. F., and HERBERT, J. E., (1989), Resource Constrained Project Crashing, *OMEGA, International Journal of Management Science*, 17(1), pp. 69–79.

De RAADT, J. D. R., (1987), Ashby's Law of Requisite Variety: An Empirical Study, *Cybernetics and Systems: An International Journal*, 18, pp. 517–36.

De SWAAN ARONS, H., (1983), Expert Systems in the Simulation Domain, *Mathematics and Computers in Simulation*, 25, pp. 10–6.

DEUTSCH, K. W., (1963), *The Nerves of Government*, Free Press: New York.

DILL, W. R., GAVER, D. P., and WEBER, W. L., (1966), Models and Modelling for Manpower Planning, *Management Science*, December, 13(4), pp. B–142–67.

DOEBLIN, E. O., (1972), *System Dynamics: Modeling and Response*, Charles E. Merrill: Columbus, Ohio.

DOOLEY, A. R., (1964), Interpretations of PERT, *Harvard Business Review*, 42(2), March–April, pp. 160–72.

DOMBROSKI, J., and SWETMAN, G., (1985), Commercially Available Knowledge–Based System Development Tools, *Proceedings of the 16th Annual Pittsburgh Conference: Modeling and Simulation*, 25–26 April, pp. 861–7.

DOUKIDIS, G. I., (1985), *Discrete Event Simulation Model Formulation Using Natural Language Understanding Systems*, Ph.D. Thesis, Department of Statistical and Mathematical Sciences, The London School of Economics.

DOUKIDIS, G. I., (1987), An Anthology on the Homology of Simulation with Artificial Intelligence, *Journal of Operational Research Society*, 38(8), August, pp. 701–12.

DOUKIDIS, G. I., and PAUL, R. J., (1985), Research into Expert Systems to Aid Simulation Model Formulation, *Journal of Operational Research Society*, 36(4), pp. 319–25.

DREYFUS, H. L., (1972), *What Computers Can't Do: A Critique of Artificial Reason*, Harper and Row: New York.

DUTTON, J. M., and STARBUCK, W. H., (1971), *Computer Simulation of Human Behaviour*, Wiley: New York.

EDWARDS, C., (1982), *Developing Micro–Computer Based Business Systems*, Prentice–Hall International.

ELLISON, D., and TUNNICLIFFE–WILSON, J. T., (1982), Interactive Simulation on a Micro–Computer, *Simulation*, May, 38(5), pp. 161–75.

ELMAGHRABY, S. E., (1967), On the Expected Duration of PERT Type Networks, *Management Science*, January, 13(5), pp. 299–306.

ELMAGHRABY, S. E., (1977), *Activity Networks: Project Planning and Control by Network Models*, Wiley: New York.

EMSHOFF, J. R. and SISSON, R. L., (1971), *Design and Use of Computer Simulation Models*, Macmillan.

FALSTER, P., (1987), Planning and Controlling Production Systems Combining Simulation and Expert Systems, *Computers in Industry*, 8, pp. 161–72.

FARNUM, N. R., and STANTON, L. W., (1987), Some Results Concerning the Estimation of Beta Distribution Parameters in PERT, *Journal of Operational Research Society*, March, 38(3), pp. 287–90.

FAURRE, P., and DEPEYPOT, M., (1970), *Elements of Systems Theory*, North–Holland: Oxford.

FEIGENBAUM, E. A., and FELDMAN, J., (1963), *Computers and Thought*, McGraw–Hill: San Francisco.

FIRTH, J. R.., (1983), *Computer Programming: Applications to Company Control and Management*, Mitchell Beasley Publishers.

FISHMAN, G. S., (1973), *Concepts and Methods in Discrete Event Digital Simulation*, Wiley Interscience: New York.

FISHMAN, G. S., (1978), *Principles of Discrete Event Simulation*, Wiley Interscience: New York.

FLITMAN, A. M., and HURRION, R. D., (1987), Linking Discrete–Event Simulation Models with Expert Systems, *Journal of Operational Research Society*, August, 38(8), pp. 723–33.

FLOOD, R. L., and CARSON, E. R., (1988), *Dealing With Complexity: An Introduction to the Theory and Application of System Science*, Plenum: New York.

FLOOD, R. L., and JACKSON, M. C., (1991), *Creative Problem Solving: Total Systems Intervention*, Wiley: Chichester

FLOOD, R. L. and STEVENS, C. G. J., (1987), Feedforward Control in Large Scale Project Management, *Proceedings of the Seventh International Congress of Cybernetics and Systems*, 9–13 September, 2, pp. 209–12, London.

FORRESTER, J. W., (1961), *Industrial Dynamics*, The MIT Press: Cambridge, Massachusetts.

FORRESTER, J. W., (1968a), *Principles of Systems*, The MIT Press: Cambridge, Massachusetts.

FORRESTER, J. W., (1968b), Industrial Dynamics—After the First Decade, *Management Science*, March, 14(7), pp. 398–415.

FORRESTER, J. W., (1968c), Industrial Dynamics—A Response to Ansoff and Slevin, *Management Science*, May, 14(9), pp. 601–18.

FORRESTER, J. W., (1969), *Urban Dynamics*, The MIT Press: Cambridge, Massachusetts.

FORRESTER, J. W., (1970), Understanding the counterintuitive behaviour of social systems, In: Beishon, J., and Peters, G., (eds.), *Systems Behaviour*, Harper and Row, pp. 270–87.

FORRESTER, J. W., (1973), *World Dynamics*, The MIT Press: Cambridge, Massachusetts.

FORRESTER, J. W., (1976), Educational Implications of Responses to System Dynamics Models, *TIMS Studies in the Management Sciences*, 2, pp. 27–35.

FORRESTER, J. W., (1987), Lessons from System Dynamics Modelling, *System Dynamics Review*, 3, pp. 136–49.

FORRESTER, J. W., LOW, G. W., and MASS, N. J., (1974), The Debate on World Dynamics: A Response to Nordhaus, *Policy Science*, 5, pp. 169–90.

FORRESTER, J. W., and SENGE, P. M., (1980), Tests for Building Confidence in System Dynamics Models, *TIMS Studies in the Management Sciences*, 14, pp. 209–28.

FOX, G. E., and BAKER, N. R., (1985), Project Selection Decision Making Linked to a Dynamic Environment, *Management Science*, October, 31(10), pp. 1272–85.

FOX, M. S., SATHI, A., and GREENBERG, M., (1984), Issues in Knowledge Representation for Project Management, *Proceedings of the IEEE Workshop on Principles of Knowledge Based Systems*, pp. 17–26.

FRANK, P. M., (1978), *Introduction to System Sensitivity Theory*, Academic Press.

FREEMAN, R. J., (1960), A Generalised PERT, *Operations Research*, 8(2), March–April, p. 281.

FRIJDA, N. H., (1967), Problems of Computer Simulation, *Behavioural Science*, 12(1), pp. 59–67.

FULKERSON, D. R., (1962), Expected Critical Path Lengths in PERT Networks, *Operations Research*, November–December, 10, pp. 808–17.

GABRIEL, E., (1986), Do Project Management Systems Provide the User with the Tools He Needs? *Proceedings of The International GPM/Internet Symposium: Project Management Software Application–Implementation–Trends*, 15–18 June, pp. 79–87, Garmisch, FRG.

GALBRAITH, J. R., (1971), Matrix Organisation Designs: How to combine functional and project forms, *Business Horizons*, February, pp. 29–40.

GALLAGHER, C., (1987), A Note on PERT Assumptions, *Management Science*, 33(10), October, p. 1360.

GAINES, B. R., (1986), Expert systems and simulation in industrial applications, *Proceedings of the Conference on Intelligent Simulation Environments—Simulation Series*, 23–25 January, 17(1), pp. 144–9, San Diego, California.

GEAR, C. W., (1971), *Numerical Initial Value Problems in Ordinary Differential Equations*, Prentice Hall .

GERALD, C. F., and WHEATLY, P. O., (1984), *Applied Numerical Analysis*, Addison–Wesley: Reading, Massachusetts.

GERNET, D., (1976), Eine Benetzernake Programmiersprache Fur Zweckedes Operations Research, *Angewandte Informatik*, 18, pp. 481–6.

GOLENKO–GINZBERG, D., (1989), PERT Assumptions Revisited, *OMEGA, International Journal of Management Science*, 17(4), pp. 393–96.

GONZALEZ, J. C., and FERNANDEZ, G., (1986), EASDM: An Expert Aid for System Dynamics Modelling, In: Aracil, J., Machuca, J. A. D., and Karsky, M., (eds.), *System Dynamics on the Move, Proceedings of the 1986 International Conference of the System Dynamics Society*, pp. 1019–27, Seville, Spain.

GOODMAN, M. R., (1974), *Study Notes in System Dynamics*, Wright–Allen: Cambridge, Massachusetts.

GORDAN, G., (1977), *System Simulation*, (2nd Ed.), Prentice–Hall: Englewood Cliffs, New Jersey.

GORDON, S., (1982), *The Dynamics of Product Development in High–Tech Firms*, MSc. Dissertation, Thayer School of Engineering, Dartmouth College, Hanover, New Hampshire.

GRAHAM, A. K., (1980), Parameter Estimation in System Dynamics Modeling, *TIMS Studies in the Management Sciences*, 14, pp. 125–42.

GREENBERGER, M., CRENSON, M. A., and CRISSEY, B. L., (1976), *Models in the Policy Process: Public Decision Making in the Computer Era*, Russell Sage Foundation: New York.

GEOFFRION, A. M., (1987), An Introduction to Structured Modeling, *Management Science*, 33(5), pp. 547–88.

GRUBBS, F. E., (1962), Attempts to Validate Certain PERT Statistics or 'Picking on PERT', *Operations Research*, November–December, 10(6), pp. 912–5.

HACKMAN, S. T., and LEACHMAN, R. C., (1989), A General Framework for Modeling Production, *Management Science*, 35(4), pp. 478–95.

HALEY, P. V., (1987), A Search Strategy for Commonsense Logic Programming, October, *Byte*, pp. 173–5.

HALPIN, D. W., (1973), *An Investigation of the Use of Simulation Networks for Modeling Construction Operations*, Ph.D. Thesis, Civil Engineering Department, University of Illinois.

HAMILTON, H. R., GOLDSTONE, S. E., MILLIMAN, J. W., PUGH, A. L., III, ROBERTS, E. B., and ZELLNER, A., (1969), *System Simulation for Regional Analysis: An Application to River Basin Planning*, The MIT Press: Cambridge, Massachusetts.

HAMMOND, K. R., (1971), Computer graphics as an aid to learning, *Science*, 172, pp. 903–8.

HAMMOND, K. R., and BREHMER, B., (1973), Quasi–rationality and distrust: implications for international conflict, In: Rapport, L., and Summers, D. A., (eds.), *Human Judgement and Social Interaction*, pp. 338–91, Holt, Rinehart and Winston: New York.

HARMON, P., and KING, D., (1985), *Expert Systems: Artificial Intelligence in Business*, Wiley: New York.

HARTLEY, H. O., and WORTHAM, A. W., (1966), A Statistical Theory for PERT Critical Path Analysis, *Management Science*, June, 12(10), pp. B–469–81.

HASTINGS, N. A. J., (1973), *Dynamic Programming with Management Applications*, Butterworths.

HEARNE, J. W., (1985), Sensitivity analysis of parameter combinations, *Applied Mathematical Modelling*, April, 9, pp. 106–8.

HEARNE, J. W., (1987), An approach to resolving the parameter sensitivity problem in system dynamics methodology, *Applied Mathematical Modelling*, August, 11, pp. 315–8.

HENRIKSEN, J. O., (1983), The Integrated Simulation Environment Simulation Software of the 1990s, *Operations Research*, November–December, 31(6), pp. 1053–73.

HIGGINS, J. C., (1985), *Computer Based Planning Systems*, Edward Arnold.

HINDELANG, T. J., and MUTH, J. F., (1979), A Dynamic Programming Algorithm for Decision CPM Networks, *Operations Research*, March–April, 27(2), pp. 225–41.

HODGETTS, R. M., (1979), *Management: Theory, Process and Practice*, W. B. Saunders Company: Philadelphia.

HOFSTADTER, D. J., (1979), *Gödel, Escher, Bach: an Eternal Golden Braid*, Penguin Books: Harmondsworth.

HOLROYD, P., MALLORY, G., PRICE, D. H. R., and SHARP, J. A., (1985), Developing Expert Systems for Management Applications, *OMEGA, The International Journal of Management Science*, 13(1), pp. 1–11.

HOLT, C. C., MODIGLIANI, F., MUTH, J. F., and SIMON, H. A., (1960), *Planning Production, Inventories and Work Force*, Prentice–Hall: Englewood Cliffs, New Jersey.

HOOPER, J. W., (1986a), Strategy–related characteristics of discrete–event languages and models, *Simulation*, April, 46(4), pp. 153–9.

HOOPER, J. W., (1986b), Activity scanning and the three–phase approach, *Simulation*, November, 46(5), pp. 210–1.

HURRION, R. D., (1978), An investigation of visual interactive simulation methods using the job–shop scheduling problem, *Journal of Operational Research Society*, November, 29(11), pp. 1085–93.

INOUE, I., and FUYUKI, M., (1987), An Expert System: Production Management Support Using Simulation –Concept, Methodology and System–, *Proceedings of the 2nd International Conference: Computer Applications in Production and Engineering*, 20–23 May 1986, 1, pp. 431–42), Copenhagen.

JACOBSON, D. H., (1968), New Second–Order and First–Order Algorithms for Determining Optimal Control: A Differential Dynamic Programming Approach, *Journal of Optimisation Theory Applications*, 12, pp. 411–40.

JACUCCI, G., and UHRIK, C., (1987), Exploring the Use of Artificial Intelligence, Logic Programming, and Computer–Aided Symbolic Manipulation in Computational Physics. 2. Expert Systems for Controlling Simulation Models, *Journal of Physical Chemistry*, 91(19), pp. 4981–9.

JANTSCH, E., (1976), Modelling the Human World: Perspectives, In: Churchman, C. W., and Mason, R. O., (eds.), *World Modeling: A Dialogue*, North–Holland/American Elsevier: New York.

JENNERGREN, L. P., (1983), Simulation in Microcomputers Revisited, *Journal of Operational Research Society*, 34(11), pp.1053–56.

JEWETT, R. F., (1967), A Minimum Risk Manpower Scheduling Technique, *Management Science*, June, 13(10), pp. B–578–92.

JOHANSON, J., (1986), More Effective Project Management With Help of a Computer, *Proceedings of The International GPM/Internet Symposium: Project Management Software Application–Implementation–Trends*, 15–18 June, pp. 103–7, Garmisch, FRG.

JOHNSON, T. J. R., (1967), *An Algorithm for the Resource Constrained Project Scheduling Problem*, Ph.D. Thesis, Alfred P. Sloan School of Management, The Massachusetts Institute of Technology, Cambridge, Massachusetts.

KAIMANN, R. A., (1974), Coefficient of Network Complexity, *Management Science*, October, 21(2), pp. 172–77.

KALYMON, B. A., (1975), An Optimization Algorithm for a Linear Model of a Simulation Model, *Management Science*, January, 21(5), pp. 516–30.

KAST, F. E., and ROSENZWEIG, J. E., (1985), *Organisation and Management: A Systems and Contingent Approach*, (4th Ed.), McGraw–Hill: Singapore.

KAVANAGH, D.P., (1985), SIREN: A Repetitive Construction Simulation Model, *Journal of Construction Engineering and Management*, September, 111(3), pp. 308–23.

KEENEY, R. L., (1982), Decision Analysis: An Overview, *Operations Research*, September–October, 30(5), pp. 803–32.

KELLEY, E. J., Jr., (1961), Critical Path Planning and Scheduling: Mathematical Basis, May–June, *Operations Research*, 9(3), pp. 296–321.

KELLEY, E. J., Jr., (1963), The Critical–Path Method: Resources Planning and Scheduling, In: Muth, J. F., and Thompson, G. L., (eds.), *Industrial Scheduling*, Prentice–Hall: Englewood Cliffs, New Jersey.

KELLY, J., (1961), Critical–Path Planning and Scheduling: Mathematical Basis, *Operations Research*, 9(3), pp. 296–320.

KELOHARJU, R., and LUOSTARINEN, A., (1983), Achieving Structural Sensitivity by Automatic Simplification, *Dynamica*, Winter, 9(Part II), pp. 60–5.

KELOHARJU, R., and WOLSTENHOLME, E. F., (1988), The Basic Concepts of System Dynamics Optimization, *Systems Practice*, 1(1), pp. 65–86.

KHOSHNEVIS, B., and AUSTIN, W. M., (1987), An intelligent interface for system dynamics modeling, Simulation and Artificial Intelligence: *Proceedings of the Conference—Simulation Series*, 18(3), pp. 81–6.

KHOSHNEVIS, B., and CHEN, A., (1987), An automated simulation modeling system based on AI techniques, Simulation and Artificial Intelligence: *Proceedings of the Conference—Simulation Series*, 18(3), pp. 87–91.

KHOSHNEVIS, B., CHEN, A., and AUSTIN, W., (1987), Intelligent Simulation Modeling Environments, *Proceedings of the 3rd Annual Artificial Intelligence and Advanced Computer Techniques Conference*, pp. 359–64.

KLAPP, O., (1986), *Overload and Boredom: Essays on the Quality of Life in the Information Society*, Greenwood Press: New York.

KLEINHANS, A., (1986), A Behaviour Analysis Expert System for System Dynamics Models, In: Aracil, J., Machuca, J. A. D., and Karsky, M., (eds.), *System Dynamics on the Move, Proceedings of the 1986 International Conference of the System Dynamics Society*, pp. 1039–47, Seville, Spain.

KLIR, G. J., and VALACH, M., (1967), *Cybernetic Modelling*, Van Nostrand: New York

KLOSGEN, W., (1986), Some Aspects of Knowledge Based Simulation and Data Analysis, *Proceedings of the 2nd European Simulation Congress*, pp. 354–60.

KNIGHT, B., and ADAMS, R., (1975), *Complex Numbers and Differential Equations*, Allen and Unwin.

KNOX, B., (1987), Project Planning, *PC User*, May, pp. 96–101.

KNUTH, D. E., (1973), Fundamental Algorithms: *The Art of Computer Programming*, Vol.1 (2nd Ed.), Addison–Wesley: Reading, Massachusetts.

KOWALSKI, R., (1974), Predicate Logic as Programming Language, In: *Proceedings of IFIP Congress 74: Information Processing*, 5–10 August, Stockholm.

KOWALSKI, R., (1979), *Logic for Problem Solving*, North–Holland, Amsterdam.

LAW, A. M., and KELTON, W. D., (1982), *Simulation Modeling and Analysis*, McGraw–Hill: New York.

LEACHMAN, R. C., and BOYSEN, J., (1985), An Aggregate Model for Multi–Project Resource Allocation, In: Dean, B. V., (ed.), *Project Management: Methods and Studies*, North Holland: New York.

LEANING, M. S., (1980), *The Validity and Validation of Mathematical Models: Methodological, Theoretical, and Practical Studies with Emphasis on the Modelling of Complex Biological Systems*, PhD Thesis, Department of Systems Science, City University.

LEE, W., and MANTEL, S. J., Jr., (1986), An Expert System for Project Termination, *Proceedings of the First International Conference on Engineering Management*, pp. 52–8.

LEGASTO, A. A., and MACIARIELLO, J., (1980), System Dynamics: A Critical Review, *TIMS Studies in the Management Sciences*, 14, pp. 23–43.

LEHMANN, A., (1987), Expert systems for interactive simulation of computer system dynamics, *Simulation and Artificial Intelligence: Proceedings of the Conference—Simulation Series*, 18(3), pp. 21–6.

LEITCH, R. R., (1987), Modelling of complex dynamic systems, *IEEE Proceedings*, 134(Part D, 4), July, pp. 245–50.

LEVIN, R. L, and KIRKPATRICK, C. A., (1979), *Quantitative Approaches to Management*, (2nd Ed.), McGraw–Hill: New York.

LEVY, F. K., THOMPSON, G. L., and WIEST, J. D., (1962), Multiship, Multishop, Workload–Smoothing Program, *Naval Research Logistics Quarterly*, 19(1), March, pp. 37–44.

LEVY, F. K., THOMPSON, G. L., and WIEST, J. D., (1963), The ABCs of the critical path method, *Harvard Business Review*, 41(5), September–October, pp. 98–108.

LINSTONE, H. A., (1983), On the Management of Technology: Old and New Perspectives, Part I, *Dynamica*, 9(2), pp. 66–73.

LINSTONE, H. A., (1984), On the Management of Technology: Old and New Perspectives, Part II, *Dynamica*, 10(1), pp. 24–33.

LITTLEFIELD, T. K., and RYAVEC, C. A., (1964), An Answer to Jasieni's Question on PERT Times, *Management Science*, 33, pp. 1357–59.

LOCK, D., (1988), *Project Management*, (4th Ed.), Gower Press: Farnborough.

LOCKYER, K. G., (1984), *Critical Path Analysis and Other Project Network Techniques*, Pitman.

LUKASZEWIEZ, J., (1965), On the estimation of errors introduced by standard assumptions concerning the distribution of activity duration in PERT calculations, *Operations Research*, 13(2), March–April, pp. 326–7.

LYNEIS, J. M., (1980) *Corporate Planning and Policy Design: A System Dynamics Approach*, The MIT Press: Cambridge, Massachusetts.

McBRIDE, R., (1985), Solving Embedded Generalized Network Problems, *European Journal of Operations Research*, 21(1), pp. 82–92.

MacCRIMMON, K. R., and RYAVEE, C. A., (1964), An Analytical Study of the PERT Assumptions, *Operations Research*, January–February, 12(1), pp. 16–37.

McCLAMROCH, N. H., (1976), A representation for multivariable feedback systems and its use in stability analysis, *International Journal of Control*, 24(1), Part 1: pp. 83–95, Part 2: pp. 97–107.

McKENNEY, J. L., (1967), Critique of: "Verification of Computer Simulation Models", *Management Science*, 14(2), October, pp. B–102–3.

McLEOD, D., (1988), SIMULATION: The formative years, *Conference Quarterly of The Society for Computer Simulation*, July, pp. 11–4.

McMILLAN, C., and GONZALEZ, R. F., (1973), *Systems Analysis: A Computer Approach to Decision Models*, (3rd Ed.), Irwin; Homewood, Illinois.

MAISEL, H., (1976), Making Management Simulation Useful, *Simulation: Simulation Today*, April, 26(4), pp. 161–4.

MALCOLM, D. G., ROSEBOOM, J. H., CLARK, C. E., and FAZAR, W., (1959), Application of a Technique for Research and Development Program Evaluation, *Operations Research*, September–October, 7(5), pp. 646–69.

MARCH, J. G., and SIMON, H. A., (1967), *Organizations*, Wiley: New York.

MARTIN, J., (1976), *Principles of Data–Base Management*, Prentice–Hall: Englewood Cliffs, New Jersey.

MASS, N. J., (ed.), (1974), *Readings in Urban Dynamics: Volume 1*, The MIT Press: Cambridge.

MASS, N. J., and SENGE, P. M., (1978), Alternative Tests for the Selection of Model Variables, *IEEE Transactions on Systems, Man, and Cybernetics*, June, SMC–8(6), pp. 450–60.

MASSEY, F. J., Jr., (1951), The Kolmogorov–Smirnov Test or Goodness of Fit, *Journal of the American Statistical Association*, March, 46(253), pp. 68–78.

MEADOWS, D. H., MEADOWS, D. L., RANDERS, J., and BEHERENS, W. W., (1972), *Limits to Growth*, Universal Books: New York.

MEREDITH, J. R., and MANTEL, S. J., (1989), *Project Management*, (2nd Ed.), John Wiley: New York.

MICHAEL, G. C., (1972), A Review of Heuristic Programming, *Decision Sciences*, 3, pp. 74–100.

MINISTRY OF DEFENCE (NAVY), (1984), *Excalibur: Submarine Refitting Project Control.* Unpublished user's guide to the use of the computerised Internet 80S PERT network scheduling system installed at H.M. Dockyard, Devonport, Plymouth.

MINSKY, M., (1982), Why People Think Computers Can't, *The AI Magazine*, Fall, 3(4), pp. 3–15.

MINTZBERG, H., (1973), *The Nature of Management Work*, Harper and Row: New York.

MITRANI, I., (1982), *Simulation Techniques for Discrete Event Systems*, Cambridge University Press: Cambridge.

MODER, J. J., PHILLIPS, C. R., and DAVIS, E. W., (1985), *Project Management with CPM, PERT and Precedence Diagramming*, (3rd Ed.), Van Nostrand: New York.

MODER, J. J., and RODGERS, E. G., (1968), Judgement Estimates of the Moments of PERT Type Distributions, *Management Science*, October, 15(2), pp. B76–83.

MONROE, A., (1962), *Digital Processes for Sampled Data Control Systems*, Wiley: New York.

MORECROFT, J. D. W., (1985), Rationality in the Analysis of Behavioural Simulation Models. *Management Science*, 31(7), July, pp. 900–16.

MOSER, J. G., (1986), Integration of artificial intelligence and simulation in a comprehensive decision–support system, *Simulation*, December, 47(6), pp. 223–9.

MOTO–OKA, T., (ed.), (1981), Fifth Generation Computer Systems: *Proceedings of the International Conference on Fifth Generation Computer Systems*, 19–22 October, Tokyo, North–Holland: Amsterdam.

MULLER–MERBACH, H., (1967), Ein Verfahren zur Planung des Optimalen Betriebsmitteleinsatzes bei der Terminierung von Grossprojekten, [A Method for the Planning of Optimal Usage of the Means of Production at the Termination of a Large Project] *Zeitschrift Fur Wirtschaftliche Fertigung*, February, Heft 2, pp. 83–8 and March, Heft 3, pp. 135–40.

MURDICK, R. G., ROSS, J. E., and CLAGGETT, J. R., (1984), *Information Systems for Modern Management*, (3rd Ed.), Prentice–Hall: Englewood Cliffs, New Jersey.

NANCE, R. E., and BALCI, O., (1987), Simulation Model Management Objectives In: Singh, M. G., (ed.), *Systems and Control Encyclopedia: Theory, Technology and Applications*, 6, pp. 4328–33.

NAYLOR, T. H., (ed.), (1979), *Simulation Models in Corporate Planning*, Praeger Publishers.

NAYLOR, T. H., BALINTFY, J. L., BURDICK, D. S., and CHU, K., (1966), *Computer Simulation Techniques*, Wiley: Chichester.

NAYLOR, T. H., and FINGER, J. M., (1967), Verification of Computer Simulation Models, *Management Science*, October, 14(2), pp. B–92–101.

NEELAMKAVIL, F., (1987), *Computer Simulation and Modelling*, Wiley: Chichester.

NEWBERRY, C., and STANDHAM, C., (1986), Parametrics—The key to Effective Project Management, *Proceedings of The International GPM/Internet Symposium: Project Management Software Application–Implementation–Trends*, 15–18 June, pp. 247–59, Garmisch, FRG.

NEWMANN, G., (1986), A Prolog Tutorial, *Proceedings of the Conference on Intelligent Simulation Environments—Simulation Series*, 23–25 January, 17(1), pp. 163–164, San Diego, California.

NIELSON, N. R., (1987), Applicability of AI techniques to simulation models, *Proceedings of the Society for Computer Simulation Simulator Conference*, pp. 35–40, Florida.

NILSSON, N. J., (1980), *Principles of Artificial Intelligence*, Tiogo Publishing: Palo Alto, California.

NORDHAUS, W., (1973), World Dynamics: Measurement Without Data, *The Economic Journal*, December, 83(332), pp. 1156–83.

OERLEMANS, T. W., TELLINGS, M. M. J., and De VRIES, H., (1972), World Dynamics: Social Feedback May Give Hope For The Future, *Nature*, August, 238, pp. 251–5.

O'KEEFE, R. M., (1984), Programming Languages, Microcomputers and O.R., *Journal of Operational Research Society*, July, 35(7), pp. 617–27.

O'KEEFE, R. M., (1986a), Simulation and expert systems—A taxonomy and some examples, *Simulation*, January, 46(1), pp. 10–6.

O'KEEFE, R. M., (1986b), The three-phase approach: A Comment on "strategy-related characteristics of discrete-event languages and models", *Simulation*, November, 47(5), pp. 208–10.

OPEN UNIVERSITY, (1975), Models and the Decision Maker, In: *Technology: A Third Level Course Systems Modelling*, T341, Unit 11, Open University Press: Milton Keynes.

OREN, T. I., (1977), Software for simulation of combined continuous and discrete systems: a state-of-the-art review, *Simulation*, February, 28(2), pp. 33–45.

OREN, T. I., (1986), Knowledge bases for an advanced simulation environment, *Proceedings of the Conference on Intelligent Simulation Environments—Simulation Series*, 23–25 January, 17(1), pp. 16–22, San Diego, California.

OREN, T. I., and ZEIGLER, B. P., (1979), Concepts for advanced simulation methodologies, *Simulation*, March, 32(3), pp. 69–82.

OZDEN, M., (1987), A Dynamic Planning Technique for Continuous Activities Under Multiple Resource Constraints, *Management Science*, 33(10), pp. 1333–47.

PALANISWAMI, S., (1987), Expert Simulation Systems for Manufacturing Planning, *Engineering Costs and Production Economics*, 11, pp. 59–62.

PARKIN, A., (1980), *Systems Management*, Edward Arnold.

PAUL, R. J., and DOUKIDIS, G. I., (1986), Further Developments in the Use of Artificial Intelligence Techniques which Formulate Simulation Problems, *Journal of Operational Research Society*, 37(8), pp. 787–810.

PAUL, R. J., and DOUKIDIS, G. I., (1987), Artificial intelligence aids in discrete-event digital simulation modelling, *IEE Proceedings*, July, 134(Part D, 4), pp. 278–86.

PEARL, J., (1984), *Heuristics: Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley: Reading, Massachusetts.

PHILLIPS, C. R., (1964), Fifteen key features of computer programs for CPM and PERT, *Journal of Industrial Engineering*, January-February, pp. 14–20.

PHILLIPS, D. T., (1974), The out-of-kilter algorithm, *Industrial Engineering*, February, 6(2), pp. 36–44.

PIDD, M., (1984), *Computer Simulation in Management Science*, Wiley: Chichester.

POPPER, K., (1959), *The Logic of Scientific Discovery*, Basic Books: New York.

PRITSKER, A. A., WATTERS, L. J., and WOLFE, D. M., (1969), Multi-Project Scheduling With Limited Resources a Zero-One Programming Approach, *Management Science*, 16(1), pp. 93–108.

PUGH, A. L., III, (1980), Integration Method: Euler or Other for System Dynamics, *TIMS Studies in the Management Sciences*, 14, pp. 179–88.

PUGH, A. L., III, (1983), *DYNAMO User's Manual*, (6th Ed.), The MIT Press: Cambridge, Massachusetts.

PUGH-ROBERTS ASSOCIATES, Inc., (1984), *User Guide and Reference Manual for Micro-DYNAMO: System Dynamics Modeling Language, IBM.PC Version*, Addison-Wesley: Reading, Massachusetts.

QUILLIAN, M. R., (1967), Word concepts: A theory and simulation of some basic semantic capabilities, *Behavioural Science*, 12(5), pp. 410–30.

RANDERS, J., (ed.), (1980), *Elements of the System Dynamics Method*, The MIT Press: Cambridge, Massachusetts.

RADER, R. J., (1968), *Advanced Software Design Techniques*, Petrocelli: New York.

RAFFERTY, T., (1985), The intelligent desk, *Computer Systems*, August, pp. 23–6.

RAO, V. R., and WEISS, E. N., (1986), On Resource Allocation Problems with Interval-Scale Coefficients, *Journal of Operational Research Society*, June, 37(6), pp. 631–5.

RATLIFF, H. D., SICILIA, G. T., and LUBORE, S. H., (1975), Finding the n Most Vital Links in Flow Networks, *Management Science*, January, 21(5), pp. 531–9.

RATTI, P., (1986), On the Use of the Artemis Language on the IBM PC for Planning and Controlling Project, *Proceedings of The International GPM/Internet Symposium: Project Management Software Application- Implementation-Trends*, 15–18 June, pp. 165–74, Garmisch, FRG.

REDDY, R., (1987), Epistemology of knowledge based simulation, *Simulation*, April, 48(4), pp. 162–6.

REICHENBACH, H., (1951), *The Rise of Scientific Philosophy*, University of California Press: Berkley.

RETTI, J., (1984), MOSES—A Frame Oriented Approach to Modelling and Simulation of Dynamic Systems, *Advances in Modelling & Simulation*, 1(4), pp. 1–10.

RICHARDSON, G. P., and PUGH, A. L., III, (1981), *Introduction to System Dynamics Modeling with DYNAMO*, The MIT Press: Cambridge, Massachusetts.

RICHMAN, E., and COLEMAN, D., (1981), Monte Carlo Simulation for Management, *California Management Review*, 25(3), pp. 82–91.

RICHMOND, B., VESCUSO, P., and PETERSON, S., (1987), *Stella for Business, (User Manual for Version 2.1)*, High Performance Systems: Lyme, New Hampshire.

RINGER, L. J., (1971), A Statistical Theory for PERT in which Completion Times of Activities are Inter–Dependent, *Management Science*, July, 17(11), pp. 717–22.

RIVETT, P., (1972), *Principles of Model Building: The Construction of Models for Decision Analysis*, Wiley: Chichester.

RIVETT, B. H. P., (1984), *Principles of Model Building*, Wiley: Chichester.

ROBERTS, E. B., (ed.), (1978), *Managerial Applications of System Dynamics*, The MIT Press: Cambridge, Massachusetts.

ROBERTS, N., ANDERSON, D. F., DEAL, R. M., GARET, M. S., and SHAFFER, W. A., (1983a), *Introduction to Computer Simulation: The System Dynamics Approach*, Addison–Wesley: Reading, Massachusetts.

ROBERTS, N., ANDERSON, D. F., DEAL, R. M., GARET, M. S., and SHAFFER, W. A., (1983b), *Instructors Manual for Introduction to Computer Simulation: The System Dynamics Approach*, Addison–Wesley: Reading, Massachusetts.

ROBERTSON, D. C., (1967), *Project Planning and Control: Simplified Critical Path Analysis*, Hetwood Books.

ROBILLARD, P., and TRAHAN, M., (1976), Expected Completion Time in PERT Networks, *Operations Research*, January–February, 24(1), pp. 177–82.

ROSENBERG, R., (1986), *Computers and the Information Society*, John Wiley and Sons: New York.

ROTHKOPF, M. H., (1973), World Models Won't Work, *Simulation*, August, 21(2), pp. 60–1.

ROZENBLIT, J. W., and ZEIGLER, B. P., (1987), *Multifaceted methodology: A framework for knowledge–based simulation modelling, Proceedings of the Conference on Methodology and Validation—Simulation Series*, 6–9 April, 19(1), pp. 50–3, January 1988, Orlando, Florida.

RUIZ–MIER, S., and TALAVAGE, J., (1987), A hybrid paradigm for modeling of complex systems, *Simulation*, April, 48(4), pp. 135–41.

RUSSELL, R. A., (1986), A Comparison of Heuristics for Scheduling Projects with Cash Flows and Resource Restrictions, *Management Science*, October, 32(10), pp. 1291–1300.

SAGE, A. P., (1981), Designs for optimal information filters, In: Nystrom, P. C., and Starbuck, W. H., (eds.), *Handbook of Organizational Design*, Vol.1, Adapting Organizations to their environments, Oxford University Press: Oxford.

SARGENT, R. G., (1984), A Tutorial on Verification and Validation of Simulation Models, *Proceedings of the 1984 Winter Simulation Conference*, 28–30 November, pp. 115–21, Dallas, Texas.

SASIENI, M. W., (1986), A Note on PERT Times, *Management Science*, 32(12), pp. 1652–3.

SASIENI, M. W., (1989), PERT Times—A Proposal for an Empirical Study, *OMEGA, International Journal of Management Science*, 17(3), p. 207.

SAYLES, L. R., (1964), *Managerial Behaviour*, McGraw–Hill: New York.

SAYER, J. S., KELLER, J. E., Jr., and WALKER, M. R., Critical Path Scheduling, *Factory*, July, pp. 74–7.

SCHARANK, W. E., and HOLT, C. C., (1967), Critique of: "Verification of Computer Simulation Models", *Management Science*, October, 14(2), pp. B–105–6.

SCHONBERGER, R. J., (1981), Why Projects Are Always Late: A rational based on manual simulation of PERT/CPM networks, *Interfaces*, 11, pp. 66–70.

SCHROEDER, W. W., III, SWEENEY, R. E., and ALFELD, L. E., (eds.), (1975), *Readings in Urban Dynamics: Volume II*, The MIT Press: Cambridge, Massachusetts.

SCHUBERT, L. K., GOEBEL, R. G., and CERCONE, N. J., (1979), The Structure and Organisation of a Semantic Net for Comprehension and Inference, In: *Associative Networks: Representation and Use of Knowledge by Computers*, Academic Press, pp. 121–75.

SCULLI, D., (1989), A Historical Note on PERT Times, *OMEGA, International Journal of Management Science*, 17(2), pp. 195–200.

SCULLI, D., and WONG, K. L., (1985), The Maximum and Sum of Two Beta Variables and the Analysis of PERT Networks, *OMEGA, International Journal of Management Science*, 13(2), pp. 233–40.

SENGE, P. M., (1977), Statistical estimation of feedback models, *Simulation*, 28(6), pp. 177–84.

SEVERIN, W., and TANKARD, J., Jr., (1987), *Communication Theories: Origins, Methods, Users*, (2nd Ed.), Pitman: New York.

SHANNON, R. E., (1975), *Systems Simulation—The Art and Science*, Prentice–Hall: Englewood Cliffs, New Jersey.

SHANNON, R. E., (1981), Tests for the Verification and Validation of Computer Simulation Models, *Proceedings of the 1981 Winter Simulation Conference*, 9–11 December, pp. 573–7, Atlanta, Georgia.

SHANNON, R. E., (1986), Intelligent simulation environments, *Proceedings of the Conference on Intelligent Simulation Environments—Simulation Series*, January, 17(1), pp. 150–6.

SHANNON, R. E., MAYER, R., and ADELSBERGER, H. H., (1985), Expert systems and simulation, *Simulation*, June, 44(6), pp. 275–84.

SHARIT, J., and SALVENDY, S., (1987), A Real–Time Interactive Model of a Flexible Manufacturing System, *IEE Transactions*, June, 19(2), pp. 167–77.

SHARP, J. A., and PRICE, S. H. R., (1982), Industry Modelling, *OMEGA, The International Journal of Management Science*, 10(3), pp. 237–47.

SHARP, J. A., and PRICE, S. H. R., (1984), System Dynamics and Operational Research: An appraisal, *European Journal of Operational Research*, 16, pp. 1–12.

SHAW, M. L. G., (1986), A framework for knowledge–based systems unifying expert systems and simulation, *Proceedings of the Conference on Intelligent Simulation Environments—Simulation Series*, January, 17(1), pp. 38–43.

SHELMERDINE, E. K., (1989), Planning for Project Management, *Journal of Systems Management*, 40, January, pp. 16–20.

SIEGEL, S., (1956), *Non–parametric Statistics for the Behavioural Sciences*, McGraw–Hill: Tokyo.

SIMON, H. A., (1957), *Administrative Behaviour*, (2nd Ed.), Macmillan: New York.

SIMON, H. A., (1960), *The New Science of Management Decision*, Harper and Row: New York

SIMON, H. A., (1969a), *The Sciences of the Artificial*, The MIT Press: Cambridge, Massachusetts.

SIMON, H. A., (1969b), *Contingencies of Reinforcement*, Appleton–Century–Crofts: New York.

SIMON, H. A., (1983), Search and Reasoning in Problem Solving, *Artificial Intelligence*, 21, pp. 7–29.

SIMON, H. A., and KADANE, J. B., (1975), Optimal Problem–Solving Search: All–or–None Solutions, *Artificial Intelligence*, 6, pp. 235–47.

SLOWINSKI, R., (1980), Two Approaches to Problems of Resource Allocation Among Project Activities—A Comparative Study, *Journal of Operational Research Society*, 31(8), August, pp. 711–23.

SMITH, A., (1983), (1st Ed. 1776), *The Wealth of Nations*, Penguin Books, Harmondsworth.

SMITH, P., (1988), *Expert System Development in Prolog and Turbo–Prolog*, Sigma Press: Wilmslow.

SPRAGUE, R. H., Jr., (1989), *Decision Support Systems: Putting theory into practice*, Prentice–Hall.

SPIEGEL, M. R., (1971), *Advanced Mathematics for Engineers & Scientists*, McGraw–Hill.

STARR, M. K., (1966), Planning Models, *Management Science*, December, 13(4), pp. B–115–41.

STARR, P. J., (1980), Modeling Issues and Decisions in System Dynamics, *TIMS Studies in the Management Science*, 14, pp. 45–59.

STAW, B. M., and ROSS, J., (1978), Commitment to a policy decision: a multi–theoretical perspective, *Administrative Science Quarterly*, 23, pp. 40–64.

STEFIK, M., AIKINS, J., BALZER, R., BENOIT, J., BIRNBAUM, L., HAYES–ROTH, F., and SACERDOTI, E., (1982), The Organisation of Expert Systems, A Tutorial, *Artificial Intelligence*, 18, pp. 135–73.

STERLING, L., (1987), Mathematical Reasoning, *Byte*, October, pp.177–80.

STERLING, L., and SHAPIRO, E., (1986), *The Art of Prolog: Advanced Programming Techniques*, The MIT Press: Cambridge, Massachusetts.

STERMAN, J. D., (1989), Modeling Managerial Behaviour: Misperceptions of Feedback in a Dynamic Decision Making Experiment. *Management Science*, 35(3), March, pp. 321–39.

STEVENS, C. G. J., (1986), *Can the Inherent Conflict of Multi–Stream Project Management be Anticipated and then Solved?* Unpublished Dissertation, Department of Systems Science, City University: London EC1V 0HB.

STEVENS, C. G. J., (1987), *From a Vote Funded Organisation to a Commercial, Employee–Owned Company: An investigation into the necessary organisational changes*, Unpublished Dissertation, Department of Systems Science, City University: London EC1V 0HB.

STEVENS, C. G. J., (1990), *Models, What Models? Do they help provide a solution or just another problem?* A presentation to the International Operational Research Society Annual Conference, Bangor.

STEVENS, C. G. J., (1991), Right Time, Right Place—Wrong Model! *Proceedings of the ISSS International Conference, 1991*, Öustersund, Sweden.

STEVENS, C. G. J., and FLOOD, R. L., (1988), *A Pragmatic Review of the Issues Involved in Development of an Artificial Intelligent Decision Support System for Use in Large Scale Projects*, Research Memorandum DSS/CGJS–RLF/251, Department of Systems Science, City University, London EC1V 0HB.

STEVENS, C. G. J., and FLOOD, R. L., (1989a), From Sequencing to 'Thinking' in Large Scale Projects, In: Flood, R. L., Jackson, M. C., and Keys, P., (eds.), *Systems Prospects: The Next Ten Years of Systems Research*, pp. 315–322. Plenum: New York.

STEVENS, C. G. J., and FLOOD, R. L., (1989b), Systemic Integration of Large Scale Project Planning Methods Into an Artificial Intelligent Decision Support System, In: Ledington, P., (ed.), *Proceedings of the ISSS International Conference, 1989*, pp. 361–6, Edinburgh.

STEVENS, C. G. J., and FLOOD, R. L., (1990), A Manager's Answer to Labour Resource Allocations in Large Scale Projects, Using an Integrated Decision Support System, In: Madisetti, V., (ed.), *Modeling and Simulation on Microcomputers, 1990*, pp. 6–11. The Society for Computer Simulation: San Diego, California.

STEVENS, C. G. J., and WHARTON, F., (1991), *Project Management*, A project management module manual for distance learning MBA, School of Management, University of Hull: Hull, Humberside HU6 7RX.

STEWART, R., (1971), *How Computers Affect Management*, Macmillan.

SWANSON, E. B., (1974), Management Information Systems: Appreciation and Involvement, *Management Science*, October, 21(2), pp. 178–88.

SWANSON, H., WOOLSEY, R. E. D., and HILLIS, H., (1974), Using the out–of–kilter algorithm, *Industrial Engineering*, March, 6(3), pp. 40–4.

STIRES, D. M.., and MURPHY, M. M., (1962), *Fundamentals of Operations Research*, Wiley: New York.

TALBOT, F. B., (1976), *An Integer Programming Algorithm for the Resource–Constrained Project Scheduling Problem*, PhD. Thesis, Pennsylvania State University.

TALBOT, F. B., (1982), Resource–Constrained Project Scheduling with Time–Resource Tradeoffs: The nonpreemptive case, *Management Science*, 28(10), October, pp. 1197–1210.

TALBOT, F. B., and PATTERSON, J. H., (1978), An Efficient Integer Programming Algorithm with Network Cuts for Solving Resource–Constrained Scheduling Problems, *Management Science*, 24(11), July, pp. 1163–74.

TANYI, E., and LINKERS, D. A., (1987), A Frame–Based Expert Systems Modelling, Colloquim on "The Use of Expert Systems in Control Engineering", Institution of Electrical Engineers: Computing and Control Division, 5th March, *Digest 1987/27*, pp. 8/1–4.

TAYLOR, B. W., and MOORE, L. J., (1980), R & D Project Planning with Q–Gert Network Modeling and Simulation, *Management Science*, January, 26(1), pp. 44–59.

THESEN, A., (1974), Some notes on systems models and modelling, *International Journal of Systems Science*, 5(2), pp. 145–52.

THIERAUF, R. J., KLEKAMP, R. C., and RUWE, M. L., (1985), *Management Sciences: A model formulation approach with computer applications*, Charles E. Merrill Publishing Company: Columbus, Ohio.

THIMM, A. L., (1974), Steps to no–fault decisions, *Journal of Systems Management*, September, pp. 25–9.

TVERSKY, A., and KAHNEMAN, D., (1974), Judgment Under Uncertainty: Heuristics and Biases, *Science*, 27th September, pp. 1124–31.

TOCHER, K. D., (1963), *The Art of Simulation*, English University Press.

TOFFLER, A., (1970), *Future Shock*, Random House: New York.

TOMLINSON, R., and KISS, I., (eds.), (1984), *Rethinking the Process of Operational Research and Systems Analysis*, Pergamon Press: Oxford.

TOMOVIC, R., and VUKOKRATOVIC, M., (1972), *General Sensitivity Theory*, North–Holland: New York.

TOWNSEND, K., (1987), Soft options for the planners, *Computing*, 10th December, p. 17.

TRANFIELD, D., (1983), Management Information Systems: An Exploration of Core Philosophies, *Journal of Applied Systems Analysis*, 10, pp. 83–9.

TRICKER, R. I., (1982), *Effective Information Management*, Beaumont Executive Press.

TURBAN, E., (1988), *Decision Support and Expert Systems: Managerial Perspectives*, Macmillan: New York.

UNWIN, A. R., and CROTTY, R., (1984), Comments on "The Development of System Dynamics as a Methodology for System Description and Qualitative Analysis", *Journal of Operational Research Society*, 35, pp. 447–8.

UYENSO, D. H., and VAESSEN, W., (1980), PASSIM: a discrete–event simulation package for Pascal, *Simulation*, December, 35(6), pp. 183–90.

UYS, P. W., (1984), An Analytical Form for Table Functions Used in System Dynamics, *Dynamica*, Winter, 10(2), pp. 81–3.

VAN SLYKE, R. M., (1963), Monte Carlo Methods and the PERT Problem, *Operations Research*, 11, pp. 839–60.

VERMEULEN, P. J., and JONGH, D. C. J., (1976), Parameter sensitivity of the 'Limits to Growth' world model, *Applied Mathematical Modelling*, 1, pp. 29–32.

WAGNER, H. M., (1969), *Dynamic Programming*, Holden–Day: San Francisco.

WAGNER, H. M., (1975), *Principles of Management Science, With Applications to Executive Decisions*, Prentice–Hall: Englewood Cliffs, New Jersey.

WALES, F. J., and LUKER, P. A., (1986), An environment for discrete event simulation, *Proceedings of the Conference on Intelligent Simulation Environments—Simulation Series*, 23–25 January, 17(1), pp. 58–62, San Diego, California.

WATERMAN, D. A., (1986), *A Guide to Expert Systems*, Addison–Wesley: Reading, Massachusetts.

WIENER, N., (1948), *Cybernetics or Control and Communications in the Animal and the Machine*, John Wiley: New York.

WEICK, K. E., (1969), *The Social Psychology of Organizing*, Addison–Wesley: Reading, Massachusetts.

WIEST, J. D., (1964), Some properties of schedules for large projects with limited resources, *Operations Research*, 12(3), May–June, pp. 395–418.

WIEST, J. D., (1967), A Heuristic Model for Scheduling Large Projects with Limited Resources, *Management Science*, February, 13(6), pp. B–359–77.

WIEST, J. D., (1981), Precedence Diagramming Method: Some Unusual Characteristics And Their Implications For Project Managers, *Journal of Operations Management*, February, 1(3), pp. 121–30.

WINOGRAD, T., (1980), Extended inference modes in reasoning by computer systems, In: Cohen, L. J., and Hesse, U., (eds.), *Application of Inductive Logic*, Clarendon Press: Oxford.

WOLSTENHOLME, E. F., (1982), System Dynamics in Perspective, *Journal of Operational Research Society*, 3, pp. 547–56.

WOLSTENHOLME, E. F., (1983), System Dynamics: A System Methodology or a System Modelling Technique, *Dynamica*, 9(2), pp. 84–90.

WOLSTENHOLME, E. F., (1990), *System Enquiry: A System Dynamics Approach*, Wiley: Chichester.

WOLSTENHOLME, E. F., and COYLE, R. G., (1980), Modelling Discrete Events in System Dynamics Models, *Proceedings of 6th International Conference on System Dynamics and Analysis of Change*, Paris.

WOLSTENHOLME, E. F., and COYLE, R. G., (1983), The Development of System Dynamics as a Methodology for System Description and Qualitative Analysis, *Journal of Operational Research Society*, 34(7), pp. 569–81.

WOODWORTH, B. M., and WILLIE, C. J., (1975), A Heuristic Algorithm for Resource Leveling in Multi–Project, Multi–Resource Scheduling, *Decision Science*, 6, pp. 525–40.

WRIGHT, R. D., (1976), Backward Integration Tests of Dynamic Models, *TIMS Studies in the Management Sciences*, 2, pp. 129–40.

WURMAN, R. S., (1991), *Information Anxiety*, Pan Books.

XIONG, G., and SONG, A., (1985), An expert system for dynamic system simulation, *Proceedings of the European Conference: AI Applied to Simulation*, pp. 106–10.

ZEIDNER, L., HAZONY, Y., and WILLIAMS, A. C., (1986), An Expert Support System to Build Expert Support Systems, *Proceedings of the International Conference: Computer Aided Design, Modelling and Simulation*, June, pp. 30–8, Vancouver.

ZEIGLER, B. P., (1976), *Theory of Modelling and Simulation*, Wiley: New York.

ZEIGLER, B. P., ELZAS, M. S., KLIR, G. J., and OREN, T. I., (eds.), (1979), *Methodology in Systems Modelling and Simulation*, North–Holland: Amsterdam.

ZEITZ, M., (1987), Simulationstechnik, [Simulation techniques], *Chemie–Ingenieur–Technik*, June, 59(6), pp. 464–9.

Commercial References

Key

ℙ => PERT System;
𝕊 => Simulation Language/Software.

BRAINPOWER PROJECT PLANNER, ℙ
Typtych,
Buckingham House,
Station Road,
Gerrards Cross,
Buckinghamshire

GENETIK, 𝕊
21 Oxford Street,
Woodstock,
Oxfordshire
OX7 1TH

GPSSR/PC, 𝕊
Simulation Software Ltd.,
760 Headley Drive,
London, Ontario
N6H 3V8 Canada

HORNET, ℙ
Claremont Controls Ltd.,
Albert House,
Rothbury, Morpeth,
Northumberland
NE65 7SR

Inter–SIM, 𝕊
Milo Consultant
Engineers Ltd.,
River View House,
London Road,
Basingstoke,
Hampshire
RG24 OJL

ISIM, 𝕊
Crosbie Hay
& Associates
P.O. Box 943,
Chico, California 95927
USA

KERNAL PROJECT MANAGEMENT SYSTEM, ℙ
United Information
Systems,
Apex House,
4A–10 West Street,
Epsom, Surrey
KT18 7RG

MAST, 𝕊
Citroen Industrie, U.K.
Automation Division,
Clarendon Avenue,
Leamington Spa,
Warwickshire
CV32 5PP

Micro PASSIM, 𝕊
Claude C. Barnett,
Walla Walla College,
College Place,
WA 99324
USA

Micro SAINT 3.0, 𝕊
Micro Analysis and
Design, Inc.,
Customer Service,
9132 Thunderhead Drive,
Boulder, Colorado 80302
USA

MICROSOFT PROJECT, ℙ
Microsoft Ltd.,
Excel House,
49 De Montfort Road,
Reading Berks.
RG1 8LP

PC–SOL, 𝕊
Systems Simulation
Consultants
11051 Ring Road,
Reston, Virginia 22090
USA

PERTMASTER, ℙ
Abtex Software,
11 Campus Road,
Listerhills Science Park,
Bradford

PLANTRAC, ℙ
Computerline Ltd.,
Tavistock House,
319 Woodham Lane,
Woodham, Weybridge,
Surrey
KT15 3PB

Professional DYNAMO PLUS, 𝕊
Alexander Pugh,
5 Lee Street,
Cambridge,
Massachusetts 02139
USA

PROJECT MANAGER WORKBENCH, ℙ
Hoskyns Group plc,
130 Shaftsbury Avenue,
London W1V 7DN

SIMNET, 𝕊
Research Computer
Services, Inc.,
1255 N. Fairfield Drive,
Beavercreek,
OH 45432
USA

SLAM II, 𝕊
Pritsker & Associates,
Inc.,
1305 Cumberland Ave.,
P.O. Box 2413,
West Lafayette,
Indiana 47906
USA

SUPERPROJECT PLUS, ℙ
Computer Associates Ltd.,
Edinburgh House,
43–51 Windsor Road,
Slough, Berks.
SL1 2EQ

TC–PROLOG, 𝕊
LOGICWARE
5000 Birch St.,
Suite 3000,
The West Tower,
Newport Beach,
California 92660
USA

TRACKSTAR, ℙ
T and B Computing, Ltd.,
19 Stratford Place,
London W1N 9AF

TURBOSIM–IV, 𝕊
Micro Simulation,
37 William J. Heights,
Framingham,
Massachusetts 01701
USA

Appendix 1

PERT and Computer Simulation Software

This appendix was originally compiled from information gathered in mid 1988 (Stevens and Flood, 1988). It reflected then, as it does now, some of the more common PERT and simulation software packages available for IBM.PC (and 100% compatible) micro-computers.

It was felt that it should be reproduced again in an Appendix to demonstrate various configurations, sizes of project that can be managed and cost of such software. Originally the research hypothesis was to extend PROMISS into an Artificial Intelligent Expert System, this, as discussed in Chapter 9 is an area for future development.

Simulation software is more diverse and complex but falls into two main categories: a) simulation shell; and b) libraries for various routines. The former are designed to be easy to use but are generally slow to execute and are more expensive to purchase. Library routines are normally written in a High Level Language and designed to help such programmers execute various simulations normally requiring a large program sub-routine.

The following tables (two related to PERT and two simulation software) contain a number of abbreviations:

- RAM Random Access Memory
- KB Kilo Byte
- MB Mega Byte
- CGA Colour Graphics Adaptor
- EGA Enhanced Graphics Adaptor

Table A1.1
PERT Software

Product name:	Plantrac	Superproject Expert	Hornet (requirements are for Hornet 5000)	Pertmaster Advance
Supplier:	Computerline Ltd Tavistock House 319 Woodham Lane Woodlam Weybridge Surrey KT15 3PB	Computer Associates Ltd Edinburgh House 43-51 Windsor Road Slough Berks SL1 2EQ	Claremont Controls Ltd Albert House Rothbury Morpeth Northumb'd NE65 7SR	Pertmaster International 8 Campus Road Listerhills Science Park Bradford W.Yorks BD7 1HR
HARDWARE REQUIREMENT				
Minimum hardware:	IBM XT or compatible	IBM XT or compatible	IBM XT or compatible	IBM XT or compatible
Recommended hardware:	IBM AT or compatible			
Minimum RAM:	256KB	512KB	256KB	320KB
Recommended RAM:	640KB	640KB	512KB	640KB
Minimum disk configuration:	10MB hard disk plus one floppy drive	10MB Hard disk plus one floppy drive	Twin floppy drives	10MB hard disk plus one floppy drive
Recommended disk configuration:			10MB hard disk plus one floppy drive	
OPERATING PARAMETERS				
Maximum project size:	250,000 activities	RAM disk space only constraint	5,000 activities	RAM disk space only constraint: (<6,000 activities and 3,000 dummies)
Maximum activities handled (inc. dummies):	250,000	1,500 with 640KB RAM	16,000+	
Maximum resources per project activity:	200	No software limit	128	No software limit
Sharing of resources, between tasks or activities?	Yes	Yes	Yes	Yes
Pricing (inc. VAT): (Education prices are not known)	1st year licence £4,025 + £3,893 for supplementary programs; subsequent years £633	£800	Hornet 2000 £1,955 Hornet 4000 £3,335 Hornet 5000 £4,085	£1,145

Table A1.2
PERT Software

Product name:	Kernal-PMS Version 5	Project Manager Workbench	Brainpower Project Planner	Microsoft Project Version 4
Supplier:	United Information Systems Apex House 4A-10 West Street Epsom Surrey KT18 7RG	Hoskyns Group plc 130 Shaftsbury Ave. London W1V 7DN	Tyrptch Buckingham House Station Road Gerrards Cross Bucks	Microsoft Ltd Excel House 49 De Monfort Rd. Reading Berks RG1 8LP
HARDWARE REQUIREMENTS				
Minimum hardware:	IBM XT or compatible	IBM PC or compatible	IBM PC or compatible	IBM PC or compatible
Recommended hardware:		IBM PC/XT/AT		
Minimum RAM:	512KB	512KB	256KB	256KB
Recommended RAM:	640KB	640KB		320KB
Minimum disk configuration:	5MB hard disk plus one floppy drive	Twin floppy drives	One floppy drive	Twin floppy drives
Recommended disk configuration:	10MB hard disk plus one floppy drive	Hard disk plus one floppy drive	Twin floppy drives	10MB hard disk plus one floppy drive
OPERATING PARAMETERS				
Maximum project size:	RAM disk space only constraints	RAM disk space only constraints	RAM disk space only constraints	RAM disk space constraints:
Maximum activities handled (inc. dummies):	5,000		1,000	200 for 256KB; 200 more for each extra 64KB up to a maximum of 999
Maximum resources per project activity:	Unlimited	200	64	8
Sharing of resources, between tasks or activities?	Yes	Yes	Yes	Yes
Pricing (inc. VAT): (Education prices are not known)	£4,025	£1,1725 per licence	£288-£863	£340 single user £685 for network

Table A1.3
Simulation Software

Product name:	TurboSim-IV	InterSim	Micro Saint	GENETIK
Supplier:	Micro Simulation 37 William J. Heights Framingham Massachusetts 01701 USA	Milo (Consultant Engineers) Ltd River View House London Road Basingstoke Hants RG24 0JL	Micro Analysis & Design 9132 Thunderhead Drive Boulder Colorado 80302 USA	Insight International Ltd 21 Oxford Street Woodstock Oxfordshire OX7 1TH
Source Code:	Yes	No	No	No
High Level Language Interface:	Turdo Pascal Turbo "C"	None	None	Not specified, but Version 5 (April 1988) is reported to have these facilities, with unspecified constraints
Artificial Intelligent Language Interface:	Turbo Prolog	None	None	
Language environment:	Turbo Pascal	ISO Pascal based on Monte Carlo methodology	None	Not specified
Accept PERT Network type simulation:	Yes	Yes	Yes	Yes
Interactive simulation possible: Device used:	Yes; Mouse and Paddles	No	Being developed. Simulation can be interrupted through menu options only	Yes; Mouse
Pricing (inc VAT on Sterling prices only):	$100 ($30 each, when 5 or more are purchased)	£863	$995	£17,710 (unspecified discount has been offerred) support 10% per annum
Hardware requirements, if known:				IBM.AT 512KB RAM

Table A1.4
Simulation Software

Product name:	MAST with SPAR and BEAM	GPSSR/PC	PC-SOL	SLAMII/PC
Supplier:	CMS Research Inc 945 Bavarian Court Oshkosh Winsconsin 54901 USA and [Citroen Industrie UK Automation Division York House Clarendon Avenue Warwicks CV32 5PP]	Simulation Software Ltd 760 Headley Drive London Ontario N6H3V8 Canada	Systems Simulation Consultants 11051 Ring Road Resham Virginia 22090 USA	Pritsker & Associates Inc 1305 Cumberland Avenue PO Box 2413 West Lafayette Indiana 47906 USA
Source Code:	Yes	No	Yes	No
High Level Language Interface:	Fortran 77 and "C" compliers	Can produce ACSII files for post-processing with other software	Turbo Pascal Version 3	Fortran and Autocard
Artificial Intelligent Language Interface:	None		Turbo Prolog	None
Language environment	Fortran 77 and "C'	None	Turbo Pascal	Not known
Accept PERT Network type simulation:	Yes - in November 1988	Yes	Yes	Yes
Interactive simulation possible: Device used:	No - IBM OS/2 under development. Interactive simulation to be introduced after this	Only to debug	No; (Would need to write a separate program code to drive it)	Yes; Mouse
Pricing (inc VAT on Sterling prices only):	£8050 £900 Education, with no training or support	$275	$500 $250 Education, ($90 for subsequent student copies)	$200 (Education) $1000 Animation system (includes SLAMII/PC)
Hardware requirements, if known:		PC 192KB RAM AT 1024KB RAM (advantage gained if an Intel 8087 floating point maths co-processor is fitted)	**Minimum:** 256KB RAM; Twin 360KB floppy disc drives; Colour or Mono Graphics Adaptor + Monitor; DOS V2.0 or up-wards **Recommended:** 640KB RAM; Twin 360 KB floppy disc drives; One hard disk + One removable hard disk is desirable; CGA + Monitor DOS V3.0 or up-wards	512KB RAM EGA + Monitor

Appendix 2

System Dynamics Subroutines

System Dynamics subroutine written in BASIC (Beginners All purpose Symbolic Instruction Code) modified to run in Microsoft BASICA on an IBM.PC/XT. This provides the program with a core subroutine to enable it to emulate one written in DYNAMO, (Roberts *et al.*, 1983b, pp. 42–3 and 47–8).

```
      SYSDYN  EDUCOMP BASIC V3.0

3000  '....SYSTEM DYNAMICS SUBROUTINE PACKAGE, GPR,
      2/4/78
3005  DIM T(22,6),X$(9),Z(6,3)
3010  READ Z4,Y8: FOR I=1 TO Z4: FOR J=1 TO 3: READ T(I,J):
      NEXT J
3012  IF T(I,3)=0 THEN PRINT "ZERO INCREMENT IN TABLE"; I:
      STOP
3015  Z=(T(I,2)-T(I,1))/T(I,3)
3017  IF Z-INT(Z)>.000001 THEN PRINT "INCREMENT ERROR IN
      TABLE"; I: STOP
3020  FOR J=4 TO Z+4: READ T(I,J): NEXT J: NEXT I
3025  READ Z1: FOR I=1 TO Z1: READ X$(I): NEXT I: Y6=T1:
      Z5=1: Y9=1
3030  PRINT "PRINT OR PLOT?";: INPUT Z$; Y$=MID(Z$,1,2): IF
      Y$="PR" THEN 3075
3035  'SCALES TO PLOT
3040  PRINT: FOR I=1 TO Z1: READ Z2(I), Z3(I): NEXT I: Z7=7:
      Z8=2: Z9=56
3045  READ W$: IF NOT (W$="OK") THEN PRINT "DATA ERROR":
      STOP
3047  DATA "OK"
3050  IF Y$="NS" THEN 3070
3055  FOR I=1 TO Z1: Z2=Z3(I)-Z2(I): PRINT X$(I); "=";
3060  FOR J=0 TO 4: PRINT TAB(Z7-Z8+J*Z9/4); Z2(I)+J*Z2/4;:
      NEXT J
3065  PRINT: NEXT I
3070  PRINT: Z3=10: RETURN
3075  'HEADINGS FOR PRINT
3080  IF Z1>6 THEN PRINT "MORE THAN 6 VARIABLES": STOP
3085  PRINT: FOR Z6=1 TO Z1: PRINT X$(Z6),: NEXT Z6: IF Z1<6
      THEN PRINT
3090  PRINT: RETURN
3100  IF T<Y6 THEN RETURN
3102  Y6=T+T3: IF Y$="PR" THEN 3200
3105  'ORDINATES
3115  FOR Z6=1 TO Z1: Y(Z6)=INT((X(Z6))/(Z3(Z6)-Z2(Z6))
      *Z9+.5): NEXT Z6
3130  'PLOT
3135  IF Z3=10 THEN PRINT T;: Z3=0
3140  PRINT TAB(Z7);: FOR Z6=0 TO Z9: FOR Z0=1 TO Z1
3145  IF Y(Z0)=Z6 THEN PRINT X$(Z0);: GOTO 3170
3150  NEXT Z0: IF 4*Z6/Z9=INT(4*Z6/Z9) THEN PRINT ".";: GOTO
      3170
3155  IF Z3>0 THEN 3165
3160  IF Z6/2=INT(Z6/2) THEN PRINT "-";: GOTO 3170
```

```
3165 PRINT ".";
3170 NEXT Z6: PRINT: 3=Z3+1: RETURN
3200 'PRINT
3205 FOR Z6=1 TO Z1: PRINT X(Z6),: NEXT Z6: IF Z1<6 THEN
     PRINT
3215 RETURN
3300 'TABLE SUBROUTINE
3305 IF X<=T(Z5,1) THEN Y=T(Z5,4): GOTO 3325
3310 IF X>=T(Z5,2) THEN Z=(T(Z5,2)-T(Z5,1))/T(Z5,3):
     Y=T(Z5,Z+4): GOTO 3325
3315 Z=INT((X-T(Z5,1))/T(Z5,3)
3320 Y=9T(Z5,Z+5)-T(Z5,Z+4))/T9Z5,3)*(X-T(Z5,1)-
     Z*T(Z5,3))+T(Z5,Z+4)
3325 Z5=Z5+1: IF Z5>Z4 THEN Z5=1
3320 RETURN
3400 'RETURN SUBROUTINE WITHOUT HIGH/LOW CAPABILITY
3405 IF X<T(Z5,1) THEN PRINT "BELOW TABLE"; Z5: "AT T="; T:
     STOP
3410 IF X>T(Z5,2) THEN PRINT "ABOVE TABLE"; Z5; "AT T="; T:
     STOP
3415 GOTO 3315
3500 'THIRD-ORDER MATERIAL DELAY SUBROUTINE
3505 Y7=Y/3: Z(Y9,0)=X*Y7: IF NOT (T=T1) THEN 3515
3510 FOR Z0=1 TO 3: Z(Y9,Z0)=Z(Y9,0): NEXT Z0: GOTO 3525
3515 FOR Z0=3 TO 1 STEP -1
3520 Z(Y9,Z0)=Z(Y9,Z0)+(T0/Y7)*(Z(Y9,Z0-1)-Z(Y9,Z0)): NEXT
     Z0
3525 Z=Z(Y9,3)/Y7: Y9=Y9+1: IF Y9>Y8 THEN Y9=1
3535 END
```

Characterisations of Variables Used in SYSDYN
(Listed Alphabetically)

I	Counters in loops: used in the subroutine beginning.
J	At line 3000 but not thereafter, so they must be used freely in the model without error.
T	Time. T varies from T1 to T2 in steps of T0.
T0	Solution interval: equivalent to DT in DYNAMO.
T1	Initial time of the simulation.
T2	Termination time of the simulation.
T3	Print or Plot period: the time elapsed between successive print statements in the PRINT mode, or between successive lines on the graph in the PLOT mode.
T*(i, j)*	The *j* th data entry in the data line for the *i* th table function.
W$	Error–detecting string: W$="OK" if data in plot–mode is "OK"; otherwise DATA ERROR is printed.
X	Input to table or third–order delay.
X*(i)*	The *i* th variable to be plotted.

X$ *(i)*	Character to be plotted for the *i* th variable.
Y	Output from a table subroutine;

ALSO

Y	Delay time in third–order delay subroutine.
Y6	"Next time to print or plot": used to signal when a print or plot period has elapsed.
Y7	One–third of the delay time in a third–order delay.
Y8	Number of third–order material delays.
Y9	Third–order delay counter: determines which delay is next, i.e., which row in the delay matrix stores the values of the levels in the pipeline for a given delay.
Y$	First two letters of Z$.
Z$	Input in response to PRINT OR PLOT?
Z *(i, j)*	The *j* th level in the *i* th third–order delay.
Z	*X* increment in a table as the table is being read;

ALSO

Z	In the table subroutines, the number of whole *X* increments (intervals) in a table to the left of the *X* input;

ALSO

Z	Output of third–order delay.
Z0	Counter in FOR/NEXT loop in plot routine and in third–order delay routine.
Z1	Number of characters to be plotted or printed.
Z2	(max *Y* – min *Y*) in setting and printing scales for the plotted variables.
Z2 *(i)*	Minimum *Y* value on scale for the *i* th variable.
Z3 *(i)*	Maximum *Y* value on scale for the *i* th variable.
Z3	Counter which determines when the time coordinate will be printed on the graph (every ten lines).
Z4	Number of table functions.
Z5	Table function counter: determines which table is next, i.e., which row of the table matrix contains the next table's data.

Z6	Counter in FOR/NEXT loops in print and plot routines.
Z7	Graph placement parameter: number of spaces from the margin to the left–hand edge of the graph.
Z8	Scale placement parameter: number of spaces the scale headings are shifted to the left to centre them over the lines of the graph to which they correspond.
Z9	Graph size parameter: width of plot. Columns number from zero through Z9, so the plot is actually Z9+1 columns wide. (If changed, Z9 should remain a multiple of four).

SYSDYN WRITTEN IN TURBO PASCAL

Below is a transposition of the above subroutine written in Turbo Pascal. PROMISS was originally based upon this but complexity necessitated it being changed. The program below is of RATPOP and compares with that produced by Roberts *et al.*, 1983b and Pugh–Roberts Associates, Inc., Reading, Massachusetts, writers of DYNAMO. Full listing is given here as an example of SD being written in another language.

```
Program Sysdyn;
Uses CRT, DOS, UTILS;
{$M 20480, 32768, 32768}{ No heap used in order to EXEC
plot.exe }
const
  Z4: integer = 1;        { No of table functions }
  Z1 = 3;                 { No of variables to print }
  T0: real = 0.25;        { Solution interval.
                            Equivalent to dt in Dynamo }
  T1: real = 0;           { Initial time of the
                            simulation }
  T2: real = 50;          { Termination time of the
                            simulation }
  T3: real = 1;           { Sampling period. Variables will
                            be printed at intervals of T3 }

var
  T: array [1..7,1..14] of real;    { T[i,j] is the jth data
                                      entry for the ith
                                      table function }
  Time: real;             { Time varies from T1 to T2 in steps
                            of T3 }
  i, j: integer;          { Counters in loops }
  Z : longint;            { x-increment in a table as a table
                            is being read }
  Y6 : real;              { Next time to print. Used to signal
                            when a print period has elapsed }
  Z5 : integer;           { Table function counter. Determines
                            which table is next, i.e which
                            row of the table matrix contains
                            the next table's data }
  Xa : array[1..Z1] of real;        { Input to table
                                      function }
  RatPop, RatFertility, SexRatio, AverageLife, Area: real;
```

```
    FemaleRatPop, RatPopDensity, InfantSurvivalMultiplier:
    real;
    RatBirthRate, AdultDeathRate: real;
    F: Text;
    TickCount: real;

  Procedure Initialise;
  begin
    T[1,1]  := 0;    T[1,2]  := 0.025; T[1,3]   := 0.0025;
    T[1,4]  := 1;    T[1,5]  := 1;     T[1,6]   := 0.96;
    T[1,7]  := 0.92; T[1,8]  := 0.82;  T[1,9]   := 0.7;
    T[1,10] := 0.52; T[1,11] := 0.34;  T[1,12]  := 0.2;
    T[1,13] := 0.14; T[1,14] := 0.1;   Y6 := T1;
    Z5 := 1;
  end;

  Procedure TableWithoutHighLowCapability(X:real; var Y:
  real);
  begin
    if X < T[Z5,1] then
      begin
        WriteLn('Below Table ', Z5,' at T=',Time);
        Halt;
      end;
    if X > T[Z5,2] then
      begin
        WriteLn('Above Table ', Z5,' at T=',Time);
        Halt;
      end;
    Z := Trunc((X-T[Z5,1])/T[Z5,3]);
    Y := (T[Z5,Z+5]-T[Z5,Z+4])/T[Z5,3]*(X-T[Z5,1]-Z*T[Z5,3])+
      T[Z5,Z+4];
    Z5 := Z5+1;
    if Z5 > Z4 then
    Z5 := 1;
  end;

  Procedure DoPrint;
  var k: integer;
  begin
    If (TickCount = 0) then
      begin
        Write(time:7:3);
        for k := 1 to Z1 do
        begin
          write(Xa[k]:12:6);
          delay(25);
        end;
        if Z1 < 6 then WriteLn;
      end;
  end;

  Procedure InitialScreen;
  var Ch: Char;
  begin
    ClrScr; CursorOff;
    GoToXY(1, 9);
    WriteLn('                    MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM;');
    WriteLn('                    :        SYSTEM DYNAMICS        :');
    WriteLn('                    :                               :');
    WriteLn('                    :     RAT POPULATION MODEL      :');
    WriteLn('                    MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM;');
    Prompt('', False);
```

```
    ClrScr; CursorSmall;
  end;

  begin
    Initialise;
    InitialScreen;
    WriteLn;
    Write('  Time      R           B          D');
    GotoXY(1, 25); ClrEol; InverseVideo;
    Write(' SPACE - pause / continue '); NormalVideo;
    Window(1,4,80,23);
    Time := T1 - T0;
    TickCount := 0;
    Assign(F,'RATPOP4.DT');
    Rewrite(F);
    repeat              { computation loop }
      if (Time = T1-T0) then
        begin
          RatPop := 10 ;
          RatFertility := 0.4;
          SexRatio := 0.5;
          AverageLife := 22;
          Area := 11000
        end
      else
        begin
          FemaleRatPop := SexRatio * RatPop;
          RatPopDensity := RatPop / Area;
          TableWithoutHighLowCapability(RatPopDensity,
          InfantSurvivalMultiplier);
          RatBirthRate := RatFertility * FemaleRatPop *
          InfantSurvivalMultiplier;
          AdultDeathRate := RatPop / AverageLife;
          Xa[1] := RatPop;
          Xa[2] := RatBirthRate;
          Xa[3] := AdultDeathRate;
          If TickCount = 0 then
            begin
              Write(F, Time);
              If Z1 >= 1 then Write(F, Xa[1]);
              If Z1 >= 2 then Write(F, Xa[2]);
              If Z1 >= 3 then WriteLn(F, Xa[3]);
            end;
          DoPrint;
          if KeyPressed then
            begin
              ch := ReadKey;
              if ch = ' ' then  Ch := ReadKey
            end;
          RatPop := RatPop + T0*(RatBirthRate-AdultDeath
          Rate);
          Tickcount := TickCount + T0;
          If TickCount = T3 then TickCount := 0;
        end;
      Time := Time + T0;
    until (Time > T2);     { End of computation loop }
    Window(1,1,80,25);
    GotoXY(1, 25); ClrEol;
    Close(F);
    Prompt('',false);
    ClrScr;
  end.
```

Appendix 3

Central Processing Unit (CPU) Specifications

The following Intel personal computer CPUs can be used in running PROMISS:

KEY	
MIPS:	Million Instructions Per Second
MBytes:	MegaBytes (10^6)
GBytes:	GigaBytes (10^8)
TBytes:	TeraBytes (10^{12})

8088
No. of Transistors: 29,000
MIPS: 0.2 (Running at 4.77MHz)
Memory Addressing:
1 MBytes

80268
No. of Transistors: 120,000
MIPS: 2.3 (Running at 12.5MHz)
Memory Addressing:
16 MBytes Physical
1 GBytes Virtual

80386 SX
No. of Transistors: 275,000
MIPS: 3 (Running at 16MHz)
Memory Addressing:
4 GBytes Physical
64 TBytes Virtual

80386 (i386)
No. of Transistors: 275,000
MIPS: 7 (Running at 33MHz)
Memory Addressing:
4 GBytes Physical
64 TBytes Virtual

80486 (i486)
No. of Transistor: 1,200,000
MIPS: 25 (Running at 33MHz)
Memory Addressing:
4 GBytes Physical
64 TBytes Virtual

CPU/BUS CONFIGURATION

8086 = 16 bit processor sitting on a 8 bit bus
8088 = 8 bit processor sitting on an 8 bit bus
80286 = 16 bit processor sitting on a 16 bit bus
80386SX = 32 bit processor sitting on a 16 bit bus
80386 = 32 bit processor sitting on a 32 bit bus
80486 = 32 bit processor sitting on a 32 bit bus (but includes an integrated numerical processor)

Appendix 4

Explanation of DYNAMO Equations for Computer Model REFIT 1/87

(Stevens, C. G. J., (1987), pp. 253–64, (developed from Stevens, C. G. J., (1986)); Richardson, G. P., and Pugh, A. L., III, (1981))

Values of "?" (Constant and Initial values) were given in original computer listing (Stevens,1987).

Refit Time

Level equation for Refit Contract Completion Date (RCCD) accumulates changes in the Number of Activities added to the Refit (NARCA):

```
NOTE    REFIT CONTRACT COMPLETION DATE
L  RCCD.K=RCCD.J+DT*NARCA.JK
N  RCCD=PRCCD
NOTE    PLANNED REFIT CONTRACT COMPLETION DATE
C  PRCCD=?
```

Remaining Time to Complete the RefiT—Planned, given its current condition, is represented by equation RTCRTP. In the computation for Required Workforce Size (RWS), time planned required can be determined from the size of the workforce and Amount of Refit Work Planned Outstanding (ARWPO):

```
NOTE    REMAINING TIME TO COMPLETE REFIT
        (PLANNED)
A  RTCRTP.K=ARWPO.K/RWS.K
```

At any one time interval from the start of the refit, Projected Refit Completion Date (PRCD) is the number of weeks, from the present, it would take to complete the refit:

```
NOTE    PROJECTED REFIT COMPLETION DATE
A  PRCD.K=TIME.K+RTCRTP.K
```

In this model it is better not to fix the Refit Contract Completion Date (RCCD) but leave it as the number of weeks from the start date. By subtracting the current value of TIME, the Project Manager can then determine Time Left to Complete the Refit (TLCR):

```
NOTE    TIME LEFT TO COMPLETE REFIT
A  TLCR.K=RCCD.K-TIME.K
```

COMPLETION OF REFIT CONTRACT

Warship/submarine refit is measured in number of activities:

```
NOTE    REFIT CONTRACT SIZE MEASURED IN
        ACTIVITIES
C   RCSMA=?
```

Planned Completion of Refit—Cumulative (PCRC) is determined by Actual Completion of Activities—Cumulative (ACAC), plus Defects Found After Test and Trials (DFATT):

```
NOTE    PLANNED COMPLETION OF REFIT
A   PCRC.K=ACAC.K+DFATT.K
```

Rate at which net additions are added to the refit contract in the future (NARCA), is the planned refit at present, minus the Projected Refit Contract Completion Date (PRCD), taking into account the time it takes to renegotiate the refit completion schedule (TRRCS) with the Fleet:

```
NOTE    NET ADDITIONS TO REFIT CONTRACT
R   NARCA.KL=(PRCD.K-RCCD.K)/TRRCS
NOTE    TIME TO RE-NEGOTIATE REFIT COMPLETION
        DATE
C   TRRCS=?
```

Level of Actual Completion of ACtivities (ACAC) at present, is determined by the ACAC previously and the progress rate between these times:

```
NOTE    ACTUAL COMPLETION (CUMULATIVE)
L   ACAC.K=ACAC.J+DT*APR.JK
N   ACAC=?
```

Amount of refit work planned at any one time as being outstanding, is determined by the Refit Contract Size Measured in Activities (RCSMA):

```
NOTE    AMOUNT OF REFIT WORK PLANNED AS
        OUTSTANDING
A   ARWPO.K=(RCSMA-PCRC.K)/PRODP.K
```

Apparent Progress Of the Refit (APOR) is not an accurate measure because it considers gross productivity (PRODG). PRODG is work completed with and without defects. All work completed is considered as Apparent Progress Of Refit (APOR) completion:

```
NOTE    APPARENT PROGRESS OF REFIT
A   APOR.K=WS.K*PRODG
NOTE    PRODUCTIVITY (GROSS)
C   PRODG=?
```

The rate Actual Progress of Refit (APR) takes into account APOR and Fraction of Work Completed Without Defects (FWCWD). Therefore, Fraction of Work Completed Without Defects (FWCWD) the ACAC divided by RCSMA:

```
NOTE    ACTUAL PROGRESS OF REFIT
R  APR.KL=APOR.K*FWCWD
NOTE    FRACTION OF WORK COMPLETED WITHOUT
        DEFECTS
C  FWCWD=?
NOTE    FRACTION OF REFIT ACTUALLY COMPLETED
A  FRAC.K=ACAC.K/RCSMA
```

WORKFORCE SIZE

Required Workforce Size (RWS) is the number of people needed to complete the refit. As the refit nears completion, the amount of activities outstanding will determine (ARWPO) against TLCR.

Due to acclimatisation time for new people being taken on for the refit, the Project Manager may be reluctant to take on additional staff—he would probably just allow the current workforce to work more overtime! Likewise, in warship/submarine refitting it is better to have the same people who stripped or removed equipment, rebuild and replace it:

```
NOTE    REQUIRED WORKFORCE SIZE
A  RWS.K=(WIWS.K*AWS.K-WIWS.K)*WS.K
```

Employment Rate (ER) in the future will be dependent upon the RWS, Actual WS and Time it takes to Redirect the Workforce (TRW):

```
NOTE    EMPLOYMENT RATE
R  ER.KL=(RWS.K-WS.K)/TRW
NOTE    TIME TO REDIRECT WORKFORCE
C  TRW=?
```

WS is a level variable influenced by new employment and Initial Size Of the Workforce (ISOW) required to undertake the refit:

```
NOTE    WORKFORCE SIZE
L  WS.K=WS.J+DT*ER.JK
N  WS=ISOW
NOTE    INITIAL SIZE OF WORKFORCE REQUIRED TO
        UNDERTAKE WARSHIP REFIT
C  ISOW=?
```

Allocated Workforce Size (AWS) from the line function is determined as ARWPO divided by TLCR:

```
NOTE    ALLOCATED WORKFORCE SIZE
A  AWS.K=ARWPO.K/TLCR.K
NOTE    WILLINGNESS TO INCREASE WORKFORCE SIZE
A  WIWS.K=TABHL(TWIWS,TLCR.K,0,35,5)
NOTE    TABLE FOR WIWS
T  TWIWS=0/0/0/0/0.25/0.8/0.95/1
```

SATISFACTORY WORK

Level of Defects Found After Test and Trials (DFATT) is an accumulation of what DFATT was before. Added to this is the Rate of Defective Work Done (RDWD) minus the rate at which it is detected by the setting to work team (Dockside Test Organisation—DTO), this is RDWFT:

```
NOTE    DEFECTS FOUND AFTER TEST AND TRIALS
L  DFATT.K=DFATT.J+DT*(RDWD.JK-RDWFT.JK)
N  DFATT=?
```

Time to detect faults, in refitted work, is the average time between the apparent completion of an activity and when it is discovered that the equipment has to be stripped down and redone. As DFATT is an average, it will become less realistic as the refit nears completion—the rate of detection will increase at the latter part of the refit, (when more tests are carried out by the DTO). Therefore, TDRW will decrease over time. ATBCT is a function of Fraction of Activities Completed Satisfactorily against that Scheduled (FACSS).

FACSS is the ratio of the number of activities believed to be completed satisfactorily against the number of activities believed required, i.e., it will vary from zero to one and as it nears one, ATBCT will decrease:

```
NOTE    AVERAGE TIME BETWEEN WORK COMPLETION
        AND TESTING
A  ATBCT.K=TABHL(TATBCT,FACSS.K,0,1,0.2)
```

TATBCT is the table function for ATBCT. As a "guesstimate" for the first half of the refit it is considered that ATBCT is fifty weeks. This will reduce to two weeks near the end of the refit:

```
NOTE    TABLE FOR ATBCT
T  TATBCT=50/40/30/15/5/2
```

Rate of Defective Work Found after Testing and trials (RDWFT) in the future is determined by DFATT and ATBCT. Most faults, when rebuilding machinery, only become known when tested:

```
NOTE    RATE OF DEFECTIVE WORK FOUND AFTER
        TESTING AND TRIALS
R  RDWFT.KL=DFATT.K/ATBCT.K
```

FACSS is the sum of PCRC and RCSMA:

```
NOTE    FRACTION OF ACTIVITIES COMPLETED
        SATISFACTORILY AGAINST THAT SCHEDULED
A  FACSS.K=PCRC.K/RCSMA
```

Fraction of Apparent Progress Of Refit (APOR) is unsatisfactory, is determined by the generation of defective work done (RDWD):

```
NOTE    RATE OF DEFECTIVE WORK DONE
R  RDWD.KL=APOR.K*(1-FWCWD)
```

The Following Are Modifications To Model REFIT 1/87 (REFIT 2/87)

(Stevens, 1987, pp. 265–72)

Refit Contract Cost

The cost at any time during of the refit is determined by the Net Charge Per Labour Week (NCLW), which at Devonport is for a 39 hour week, and the Effect of Overtime Working—Cumulative (EOTWC):

```
NOTE    REFIT CONTRACT COST
A  RCC.K=NCLW*EOTWC.K
NOTE    NET CHARGE PER LABOUR WEEK
C  NCLW=?
```

Refit Completion

To make the Current Refit Contract Size Measured in Activities (CRCSMA) reflect that normally found in a normal submarine refit, a table function is used. Final Refit Contract Size Measured in Activities (FRCSMA) also follows the same table. This table function works out a ratio so that when the refit is completed (zero activities remaining) the simulation will then stop:

```
NOTE    CURRENT REFIT CONTRACT SIZE MEASURED IN
        ACTIVITIES
A  CRCSMA.K=TABHL(TCRCSA,FRAC.K,0,1,0.2)
NOTE    TABLE FOR CRCSMA
T  TCRCSA=2500/3750/4750/5000/5050/5250
```

and:

```
NOTE    FINAL REFIT CONTRACT SIZE MEASURED IN
        ACTIVITIES
N  FRCSMA=TABLE(TCRCSA,1,0,1,0.2)
```

Apparent Progress Of the Refit (APOR) has been changed to take into account the Effect on Gross Productivity to Meet a tight Completion date (EGPMC). This is explained further below:

```
NOTE    APPARENT PROGRESS OF REFIT
A  APOR.K=WS.K*PRODG.K*EGPMC.K
```

WORKFORCE SIZE

Workforce Size (WS), in this series of refit models, is made up of both suitably trained and experienced plus the number of casual people taken on just for the refit:

```
NOTE    WORKFORCE SIZE
A  WS.K=SWSTE.K+CWFRC.K
```

The level for Size of Workforce Suitably Trained and Experienced (SWSTE) takes into account the Acclimatising Rate for Casual Workforce (ARCW):

```
NOTE    SIZE OF WORKFORCE SUITABLY TRAINED AND
        EXPERIENCED
L  SWSTE.K=SWSTE.J+DT*ARCW.JK
N  SWSTE=ISWSTE
NOTE    INITIAL VALUE OF SWSTE
C  ISWSTE=?
```

Casual Workforce taken on For the Refit Contract (CWFRC) is a level considering the Employment Rate (ER) minus ARCW over the previous refit period:

```
NOTE    CASUAL WORKFORCE TAKEN ON FOR REFIT
        CONTRACT
L  CWFRC.K=CWFRC.J+DT*(ER.JK-ARCW.JK)
N  CWFRC=IVOCWF
NOTE    INITIAL VALUE OF CWFRC
C  IVOCWF=?
```

When taking on a casual workforce (in part at least) then (ARCW) for the future must be calculated. This rate considers CWFRC and Time To Acclimatise Casual Workforce (TTACW):

```
NOTE    ACCLIMATISING RATE FOR CASUAL WORKFORCE
R  ARCW.KL=CWFRC.K/TTACW
NOTE    TIME TO ACCLIMATISE CASUAL WORKFORCE
C  TTACW=?
```

The more experienced the workforce, the more work will be completed without defects. Effect of Training and Experience on the Fraction of Work Completed without Defects (ETEFCD) is therefore a function of Fraction of Workforce Suitably Trained and Experienced (FWSTE). This is SWSTE, calculated as a fraction of WS:

```
NOTE    EFFECT OF TRAINING AND EXPERIENCE ON
        FRACTION OF WORK COMPLETED WITHOUT DEFECTS
A  ETEFCD.K=TABHL(TETFCD,FWSTE.K,0,1,0.2)
NOTE    TABLE FOR ETEFCD
T  TETFCD=0.5/0.6/0.75/0.8/0.95/1
NOTE    FRACTION OF WORKFORCE SUITABLY TRAINED
        AND EXPERIENCED
A  FWSTE.K=SWSTE.K/WS.K
```

Effect of a trained and Experienced Workforce on Gross Productivity (EEWGP) is also considered as a function of FWSTE. CWFRC will affect FWSTE because they have to help the CWFRC until they are competent enough to work on their own, and Productivity returns to normal (PRODGN):

```
NOTE    EFFECT OF A TRAINED AND EXPERIENCED
        WORKFORCE ONGROSS PRODUCTIVITY
A  EEWGP.K=TABHL(TEEWGP,FWSTE.K,0,1,0.125)
NOTE    TABLE FOR EEWGP
T  TEEWGP=0.3/0.35/0.4/0.6/0.65/0.75/0.8/0.85/1
NOTE    PRODUCTIVITY (GROSS)
A  PRODG.K=PRODGN*EEWGP.K
C  PRODGN=?
```

Fraction of Work Completed Without Defects (FWCWD) will be a function of Normal Fraction of Work Completed (FWCWD), ETEFCD and the Effect of a tight Completion date on Work Completed without Defects (ECWCD):

```
NOTE    FRACTION OF WORK COMPLETED WITHOUT
        DEFECTS
A  FWCWD.K=NFWCWD.K*ETEFCD.K*ECWCD.K
```

Normal Fraction of Work Completed Without Defects (NFWCWD) is a function of Fraction of Refit Actually Completed (FRAC):

```
NOTE    NORMAL FRACTION OF WORK COMPLETED
A  NFWCWD.K=TABHL(TNFWCD,FRAC.K,0,1,0.125)
NOTE    TABLE FOR NFWCWD
T  TNFWCD=0.5/0.55/0.6/0.65/0.7/0.8/0.85/0.9/1
```

EGPMC is likely to improve as pressure is put on the workforce as completion date nears. However, as the workforce is put under this pressure, they are more likely to make mistakes. Therefore, ECWCD will also have a similar function as EGPMC:

```
NOTE    EFFECT ON GROSS PRODUCTIVITY TO MEET
        A TIGHT COMPLETION DATE
A  EGPMC.K=TABHL(TEGPMC,PRCD.K/RCCD.K,0.5,0.8,
   0.05)
NOTE    TABLE FOR EGPMC
T  TEGPMC=0.90.95/1/1.05/1.1/1.2/1.25
NOTE    EFFECT OF A TIGHT COMPLETION DATE ON
        WORK COMPLETED WITHOUT DEFECT
A  ECWCD.K=TABLE(TECWCD,PRCD.K/RCCD.K,0.9,
   1.2,0.05)
NOTE    TABLE FOR ECWCD
T  TECWCD=1.1/1/0.9/0.8/0.75/0.7/0.65
```

EOTWC will be determined by the WS and EGPMC. Overtime is also likely to increase as the completion date nears and the Project Manager takes on less staff (WIWS):

```
NOTE    EFFECT OF OVERTIME WORKING (CUMULATIVE)
L  EOTWC.K=EOTWC.J+DT*(WS.J*EGPMC.J)
N  EOTWC=?
```

PRODUCTIVITY

PRODG is unsatisfactory plus satisfactory work;

PRODA is only satisfactory work;

PRODP is projected from historical data of previous refits and pitched between PRODG and PRODA;

PRODI is the weighted productivity average.

Productivity—Indicated (PRODI), is the weighted average between Productivity—Actual (PRODA), which is work satisfactorily completed, and the Productivity—Planned (PRODP).

Weighting factor WGAP is formulated so it moves from 0 → 1 as the refit nears completion. As the reporting system gains more accurate information of refit progress, less emphasis is given to PRODI. This weighting is worked out with a function table of FACSS. Due to the delay in adjusting PRODP indicator, it is necessary to SMOOTH the weighting average:

```
NOTE    PRODUCTIVITY (INDICATED)
A  PRODI.K=WGAP.K*PRODA.K+(1-WGAP.K)*PRODG
NOTE    PRODUCTIVITY (ACTUAL)
A  PRODA.K=PRODG*FWCWD
NOTE    WEIGHT GIVEN TO ACTUAL PRODUCTIVITY
A  WGAP.K=TABHL(TWGAP,FACSS.K,0,1,0.125)
NOTE    TABLE FOR WGAP
T  TWGAP=0/0.05/0.15/0.25/0.45/0.65/0.85/0.9/1
NOTE    PRODUCTIVITY (PLANNED)
A   PRODP.K=SMOOTH(PRODI.K,TBAPD)
NOTE    TIME TO BECOME AWARE OF PRODUCTIVITY
C   TBAPD=?
```

ABBREVIATIONS USED IN REFIT/87 SERIES OF COMPUTER SIMULATIONS

– A –

ACAC = Actual Completion of Activities—Cumulative.

APR = Actual Progress of Refit.

APOR = Apparent Progress Of Refit.

ARCW = Acclimatising Rate for Casual Workforce.

ARWPO = Amount of Refit Work Planned Outstanding.

ATBCT = Average Time Between work Completion and Testing.

AWS = Allocated Workforce Size.

– C –

CRCSMA = Current Refit Contract Size Measured in Activities.

CWFRC = Casual WorkForce taken on for Refit Contract.

– D –

DFATT = Defects Found After Test and Trials.

– E –

ECWCD = Effect of a tight Completion date on Work Completed without Defect.

EEWGP = Effect of a trained and Experienced Workforce on Gross Productivity.

EGPMC = Effect on Gross Productivity to Meet a tight Completion date.

EOTWC = Effect of Overtime Working—Cumulative.

ER = Employment Rate.

ETEFCD = Effect of Training and Experience on Fraction Completed without Defects.

– F –

FACSS = Fraction of Activities Completed Satisfactorily against that Scheduled.

FRAC = Fraction of Refit Actually Completed.

FRCSMA = Final Refit Contract Size Measured in Activities.

FWCWD = Fraction of Work Completed Without Defects.

FWSTE = Fraction of Workforce Suitably Trained and Experienced.

– I –

IRCCD = Initial Refit Contract Completion Date.

ISWSTE = Initial Size of Workforce Suitably Trained and Experienced.

IVOCWF = Initial Value Of Casual Workforce.

– N –

NARCA = Net Additions to Refit ContrAct (Manweeks/Week).

NCLW = Net Charge per Labour Week.

NFWCWD = Normal Fraction of Work Completed Without Defects.

– P –

PCRC = Planned Completion of Refit—Cumulative.

PRCD = Projected Refit Completion Date.

PRODA = PRODuctivity (Actual).

PRODG = PRODuctivity (Gross).

PRODGN = PRODuctivity (Gross, Normal).

PRODI = PRODuctivity (Indicated).

PRODP = PRODuctivity (Planned).

– R –

RCC = Refit Contract Cost.

RCCD = Refit Contract Completion Date.

RDWD = Rate of Defective Work Done.

RDWFT = Rate of Defective Work Found after Testing and trials.

RTCRTP = Remaining Time to Complete RefiT (Planned).

RWS = Required Workforce Size.

– S –

SWSTE = Size of Workforce Suitably Trained and Experienced.

– T –

TATCT = Table for Average Time between work Completion and Testing.

TBAPD = Time to Become Aware of ProDuctivity.

TCRCSA = Table for Current Refit Contract Size measured in Activities.

TECWCD = Table for Effect of a tight Completion date on Work Completed without Defects.

TEEWGP = Table for Effect of a trained and Experienced Workforce on Gross Productivity.

TEGPMC = Table for Effect on Gross Productivity to Meet a tight Completion date.

TETFCD = Table for Effect of Training and experience on Fraction of work Completed without Defects.

TLCR = Time Left to Completed Refit.

TNFWCD = Table for Normal Fraction of Work Completed without Defects.

TRRCS = Time to Renegotiate Refit Completion Schedule.

TRW = Time to Redirect Workforce.

TTACW = Time To Acclimatise Casual Workforce.

TWGAP = Table for Weighting Given to Actual Productivity.

TWIWS = Table for Willingness to Increase Workforce Size.

– W –

WGAP = Weighting Given to Actual Productivity.

WIWS	=	Willingness to Increase Workforce Size.
WS	=	Workforce Size.

Types of Equations Used In Micro–DYNAMO

(Pugh, 1983; Pugh–Roberts Associates, 1984))

A = AUXILIARY
C = CONSTANT, GIVEN
L = LEVEL
N = INITIAL VALUE or CONSTANT, COMPUTED
R = RATE
T = TABLE and TABHL STATEMENTS (see below)

DIMENSIONLESS = Having no units associated with a quantity. For example, the ratio of two quantities with the same units is dimensionless.

TABLE function = A DYNAMO algorithm for expressing a graphical relationship, usually non–linear, between two or more variables. The form of the table look–up function is:

TABLE (TAB, X, XLOW, XHIGH, XINCR)

where:

TAB – is the name of the table;
X – is the independent variable;
XLOW – is the lowest value of the range of the independent variable;
XHIGH – is the highest value of the range of the independent variable.

TABHL function = Same as TABLE function, but permits independent variable to exceed the upper limit specified. The format is:

TABHL (TAB, X, XLOW, XHIGH, XINCR)

where:

TAB – is the name of the table;
X – Is the independent variable;
XLOW – is the lowest value of the range of the independent variable;
XHIGH – is the highest value of the range of the independent variable;
XINCR – is the increment between values of the independent variable.

SMOOTH function = A DYNAMO function that performs a weighted averaging of past values, using an exponential average.

PLOT PERIOD = The interval between successive entries in an output graph. In DYNAMO, the plot period is set by giving the constant PLTPER a value.

PRINT PERIOD	=	The time interval between successive printing of output values. In DYNAMO, the print period is set by giving the constant PRTPER a value.
TIMESCRIPTS	=	A shorthand way of representing a time interval. In System Dynamics, .J, .K and .L are used to represent past, present and future time, respectively.

Appendix 5

User's Guide to PROMISS

This Appendix briefly demonstrates, to a user, what screens will appear when certain courses of action are taken.

PROMISS is a large program and only able to run on a PC (the faster the better) with or without a numerical coprocessor. Before installing PROMISS for the first time users must read the READ.ME file (just type TYPE READ.ME <RETURN>).

PROMISS requires the following hardware configuration:

- IBM PC/XT or 100% compatible running under PC/MS-DOS version 2.1 or above;
- A numeric coprocessor;
- A minimum of 2 Mb of free disk space after PROMISS has been installed (more if a large number of data files are generated);
- 640 Kb of RAM (Unload all memory resident programs (TSRs) e.g. SideKick, or else PROMISS will have difficulties in running).

Preferred configuration: 25Mhz 80386 with 5 Mb Free Disk space

NOTES:

- Type INSTALL at the DOS prompt for an automatic installation of PROMISS;
- Users with an IQ less than 148 should leave the program alone!

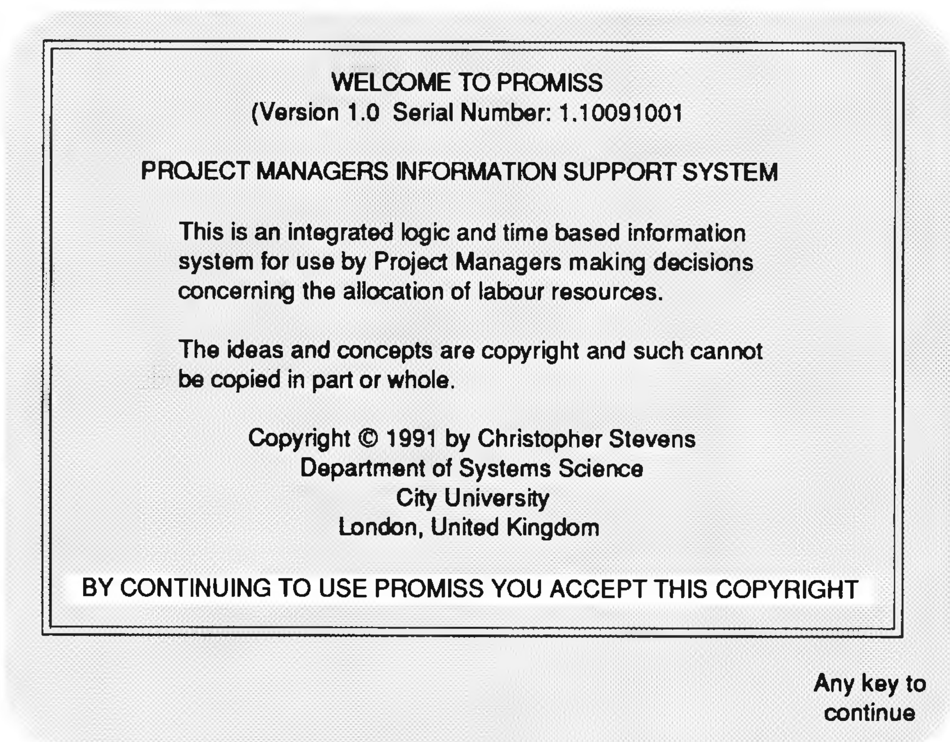
NOTE: whilst many operations will start on any key, some request specific keys i.e., inside < > will refer to that specific key e.g., <RETURN> means the return key.

PROMISS will be automatically installed once INSTALL and <RETURN> is typed. The program will be installed in its own directory named PROMISS and files relating to this program should be kept inside it (the program will provide the user with a selection menu, specific to any operation, as and when required). Once installed, PROMISS will start after PROMISS <RETURN> is typed. Future users will require the directory PROMISS to opened before starting. For example if PROMISS is kept on disc C: the following should be typed against the prompt:

C:\> cd promiss <RETURN>
Then type PROMISS <RETURN> to start.

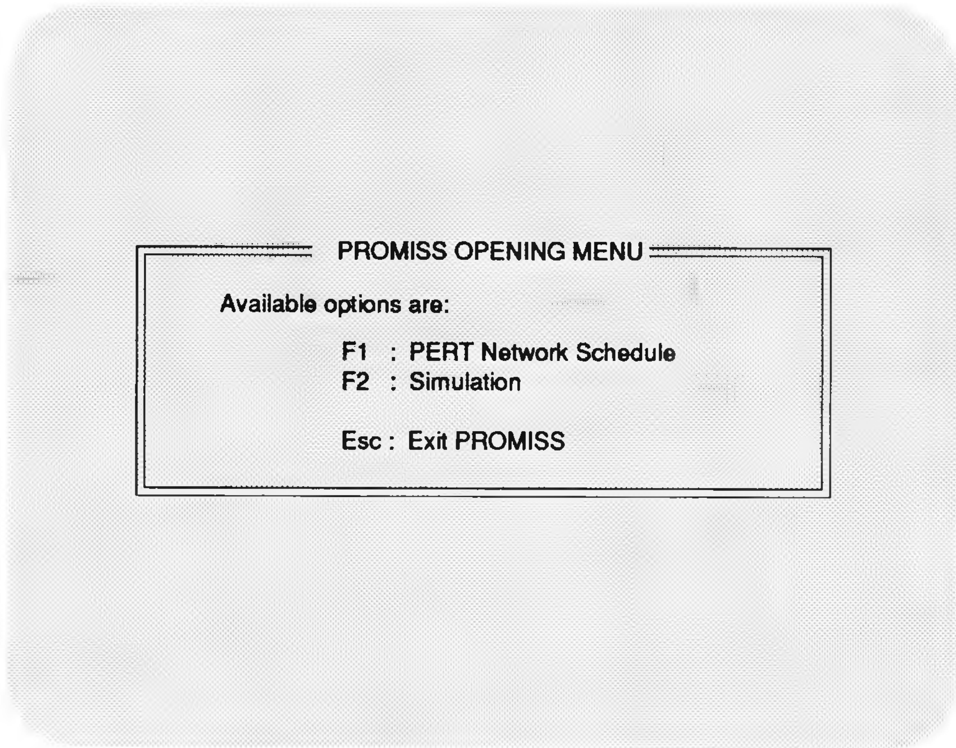
To gain maximum benefit from PROMISS it should be used by someone who is familiar with fundamental concepts of a PERT network schedule.

Once the user has started PROMISS he will be presented with an opening screen. Fundamental concepts of PROMISS (Copyright © 1991 by C. G. J. Stevens) are copyright and legal action *will* be taken if used without written permission of the author. Further use is only to be undertaken once this copyright is accepted, users must not reverse engineer it.



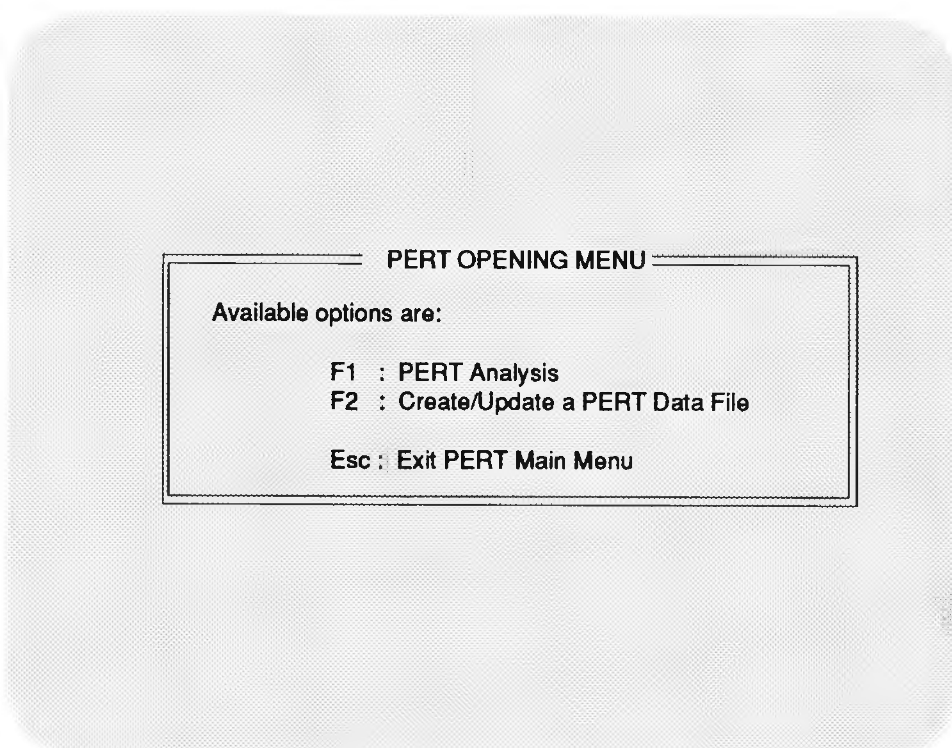
Users can access PROMISS Opening Menu by using any <KEY>. Screen 1 will present the user with an option to enter the PERT network schedule or simulation mode. When starting with a new project, PERT must be used first—this provides the simulation model with an initial database from which to work. If this has already been created, then use can be made of the simulation model. Should a user access the simulation model part of PROMISS by mistake, he can return to the Main Menu by using <ESC> and then enter the

PERT. In general terms, assess down through the menu structure is through the use of function keys and return is by use of <ESC>.



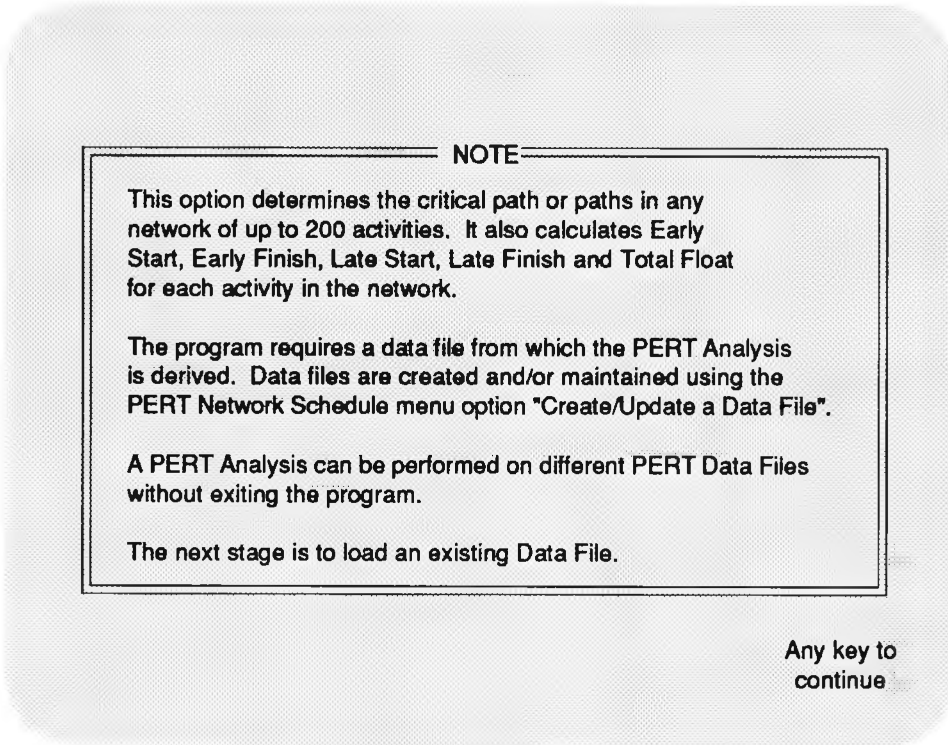
Screen 1

Screen 2 is the PERT opening menu. A user can either analyse or update an existing file or create a new one.



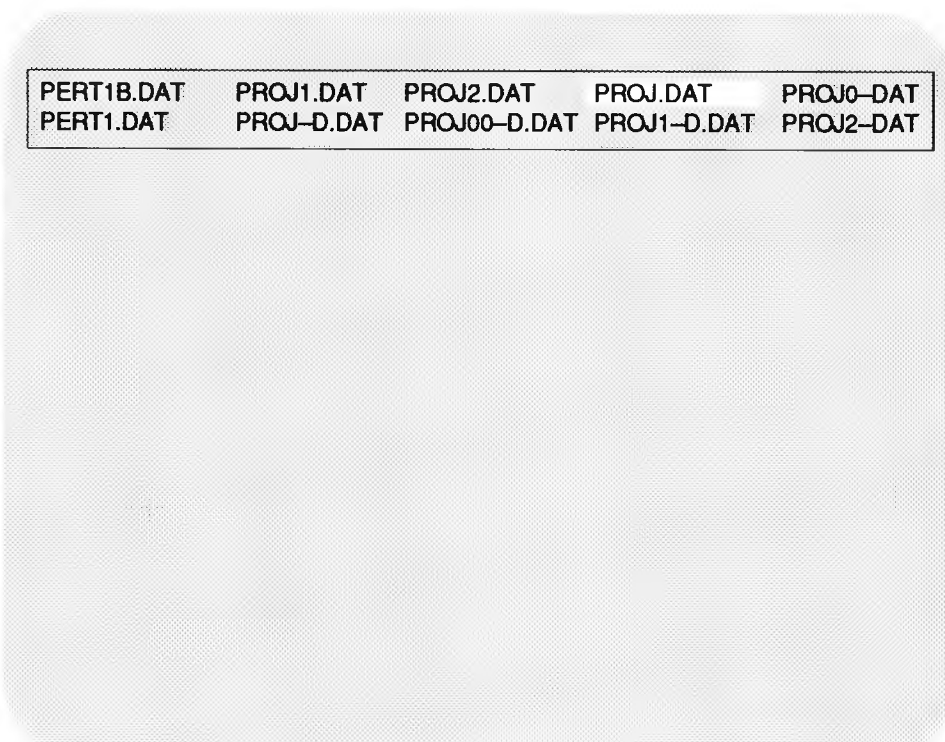
Screen 2

By selecting <F1> the user is reminded of PROMISS' PERT limitations. As a development program two hundred activities is the maximum sized network.



Screen 3

Screen 4 provides a data file menu listing all existing PERT files available for analysis. Selection is by moving a reverse video box around the screen with cursor keys, <RETURN> will load a file.



Screen 4

Once a file has been selected users are, in Screen 5, asked for the time units to be used

```
Time unit to use: (H)our (D)ay or (W)eek? _
```

Screen 5

Screen 6 and 7 show the listing of a file giving details of activity number and name, preceding activities, times and variances.

Act No	Act Name	Preceding activities			Expected Time (D)	Variance
1	1–2				1.00	0.00
2	2–3	1			1.25	0.00
3	2–4	1			0.25	0.00
4	3–4	2			0.50	0.00
5	3–6	2			0.63	0.00
6	4–6	3			1.50	0.00
7	4–7	3			1.00	0.00
8	5–8	4			0.38	0.00
9	6–7	5	6		1.00	0.00
10	6–8	5	6		0.25	0.00
11	7–8	7	9		0.88	0.00
12	8–9	8	10	11	0.50	0.00
13	11–12				1.38	0.00
14	12–13	13			0.75	0.00
15	12–14	13			1.25	0.00
16	13–15	14			1.00	0.00
17	13–16	14			1.00	0.00

SPACE – pause / continue

Screen 6

Act No	Act Name	Preceding activities			Expected Time (D)	Variance
6	4–6	3			1.50	0.00
7	4–7	3			1.00	0.00
8	5–8	4			0.38	0.00
9	6–7	5	6		1.00	0.00
10	6–8	5	6		0.25	0.00
11	7–8	7	9		0.88	0.00
12	8–9	8	10	11	0.50	0.00
13	11–12				1.38	0.00
14	12–13	13			0.75	0.00
15	12–14	13			1.25	0.00
16	13–15	14			1.00	0.00
17	13–16	14			1.00	0.00
18	14–16	15			2.00	0.00
19	14–17	15			0.25	0.00
20	15–18	16			1.38	0.00
21	16–17	17	18		0.25	0.00
22	16–18	17	18		0.75	0.00
23	17–18	19	21		1.50	0.00
24	18–19	20	22	23	0.63	0.00

*** End of File ***

Any key to continue

Screen 7

After this listing it is necessary to perform a PERT analysis from Screen 8. An alternative, at this point, is to load a different data file.

PERT ANALYSIS MENU

F1 : Perform PERT Analysis on current PERT Data File
F2 : Load a Different PERT Data File

Esc : Exit PERT Analysis Menu

Current Data File: **PROJ.DAT**

Screen 8

Screen 9 asks the user to wait whilst the PERT analysis is in progress.

PERT Analysis in progress. Please Wait . . _

Screen 9

Once the analysis calculations have been performed the user can analyse a PERT network schedule, print out results, modify activities and level resources. Screen 10 provides a list of these options.

PERT PRINT AND LEVELING MENU

Available options are:

F1 : Print Critical Paths
F2 : Print all Activity Starts, Finishes and Floats
F3 : Produce a Daily Project Work Table
F4 : Level Resources
F5 : Print a Resource Leveling Results File
F6 : Modify an Activity
F7 : Activity Print Menu
TAB : Toggle Hard Copy Mode; Currently DISABLED

Esc : Exit Print and Leveling Menu

Current Data File: **PROJ.DAT**

Screen 10

On selecting <F1> and choosing a time unit activities on the Critical Path are printed as in Screen 11 and 12. When file has reached the end Project Length will be given. Final activity(s) of the network schedule is highlighted.

Results File: **PROJ.PRT**

Critical Path:

Activity Number	Activity Name
13	11–12
15	12–14
18	14–16

SPACE – pause / continue

Screen 11

Results File: **PROJ.PRT**

Critical Path:

Activity Number	Activity Name
13	11–12
15	12–14
18	14–16
21	16–17
23	17–18
24	**18–19**

Project Length: 7.00 Days
Final activity to be completed is highlighted

Any key to continue

Screen 12

After returning to the PERT PRINT AND LEVELLING MENU and then selecting <F2> the user, having chosen the data file and time units will be able to view various start/finish times and floats. Screen 13 is in "paused" mode and Screen 14 when end of file is reached.

Act No	Activity Name	Length (D)	Variance	Early Start	Early Finish	Late Start	Late Finish	Total Float	
1	1–2	1.00	0.00	0.00	1.00	1.75	2.75	1.75	
2	2–3	1.25	0.00	1.00	2.25	2.75	4.00	1.75	
3	2–4	0.25	0.00	1.00	1.25	2.88	3.13	1.88	
4	3–4	0.50	0.00	2.25	2.75	5.63	6.13	3.38	
5	3–6	0.63	0.00	2.25	2.88	4.00	4.63	1.75	
6	4–6	1.50	0.00	1.25	2.75	3.13	4.63	1.88	
7	4–7	1.00	0.00	1.25	2.25	4.63	5.63	3.38	
8	5–8	0.38	0.00	2.75	3.13	6.13	6.50	3.38	
9	6–7	1.00	0.00	2.88	3.88	4.63	5.63	1.75	
10	6–8	0.25	0.00	2.88	3.13	6.25	6.50	3.38	
11	7–8	0.88	0.00	3.88	4.75	5.63	6.50	1.75	
12	8–9	0.50	0.00	4.75	5.25	6.50	7.00	1.75	
13	11–12	1.38	0.00	0.00	1.38	0.00	1.38	0.00	Cr
14	12–13	0.75	0.00	1.38	2.13	2.88	3.63	1.50	

SPACE – pause / continue Cr : Critical

Screen 13

Results File: **PROJ.PRT**

Act No	Activity Name	Length (D)	Variance	Early Start	Early Finish	Late Start	Late Finish	Total Float	
10	6–8	0.25	0.00	2.88	3.13	6.25	6.50	3.38	
11	7–8	0.88	0.00	3.88	4.75	5.63	6.50	1.75	
12	8–9	0.50	0.00	4.75	5.25	6.50	7.00	1.75	
13	11–12	1.38	0.00	0.00	1.38	0.00	1.38	0.00	Cr
14	12–13	0.75	0.00	1.38	2.13	2.88	3.63	1.50	
15	12–14	1.25	0.00	1.38	2.63	1.38	2.63	0.00	Cr
16	13–15	1.00	0.00	2.13	3.13	4.00	5.00	1.88	
17	13–16	1.00	0.00	2.13	3.13	3.63	4.63	1.50	
18	14–16	2.00	0.00	2.63	4.63	2.63	4.63	0.00	Cr
19	14–17	0.25	0.00	2.63	2.88	4.63	4.88	2.00	
20	15–18	1.38	0.00	3.13	4.50	5.00	6.38	1.88	
21	16–17	0.25	0.00	4.63	4.88	4.63	4.88	0.00	Cr
22	16–18	0.75	0.00	4.63	5.38	5.63	6.38	1.00	
23	17–18	1.50	0.00	4.88	6.38	4.88	6.38	0.00	Cr
24	**18–19**	**0.63**	**0.00**	**6.38**	**7.00**	**6.38**	**7.00**	**0.00**	**Cr**

Project Length: 7.00 Days
Final activity to be completed is highlighted

Any key to continue

Screen 14

From PERT PRINT AND LEVELLING MENU <F3> is selected with a data and time units to provide either a (D)aily or (W)eekly printout of resource allocations for each Labour Type, see Screen 15. Screen 15 is after "Pause" and Screen 16 at the end of file also provides Project Length together with Total Resource Allocations.

Results File: PROJ.PRT

Week,Day	MF	EF	CT	PT	GL	Man Days
1,1	5.00	0.00	0.00	0.00	0.00	5.00
1,2	6.13	4.00	0.00	0.00	0.00	10.13
1,3	9.00	8.88	0.00	0.00	0.00	17.88
1,4	3.63	4.00	0.00	0.00	0.00	7.63
1,5	3.25	3.25	0.00	0.00	0.00	6.50
2,1	4.00	1.63	0.00	0.00	0.00	5.63

SPACE – pause / continue ESC – Exit

Screen 15

Results File: PROJ.PRT

Week.Day	MF	EF	CT	PT	GL	Man Days
1.1	5.00	0.00	0.00	0.00	0.00	5.00
1.2	6.13	4.00	0.00	0.00	0.00	10.13
1.3	9.00	8.88	0.00	0.00	0.00	17.88
1.4	3.63	4.00	0.00	0.00	0.00	7.63
1.5	3.25	3.25	0.00	0.00	0.00	6.50
2.1	4.00	1.63	0.00	0.00	0.00	5.63
2.2	1.50	2.50	0.00	0.00	0.00	4.00

Project Length: 7.00 Days Total Man Days: 56.75

Results saved in daily table: **PROJ.TAB**

Any key to continue

Screen 16

When selecting <F4> from PERT PRINT AND LEVELLING MENU the levelling procedure is started. Screen 17 draws the user's attention to limitations of the levelling heuristic. Before levelling starts the user can vet resource allocations individually against activities. This information is shown in Screen 18, 19 and 20. Screen 21 provides a summary of these results.

NOTE

Due to the heuristic nature of the leveling procedure, the algorithm used here cannot guarantee the optimum schedule. It is however accepted that there is a high, although unknown, probability of being close to the optimum.
The present configuration requires the user to assign priority to a particular resource type. Activities requiring this resource will be optimised with regards to this priority.

The next 5 screens show the resource assignment per activity for the whole project and should help the user decide what resource type is to be given the highest priority (e.g. Most Expensive, Shortest in Supply, etc.).

Any key to continue

Screen 17

Activity Number	Activity Name	Elec Fitters Allocated	Critical Activity
3	2–4	5	
4	3–4	4	
6	4–6	2	
8	5–8	2	
10	6–8	5	
12	8–9	2	
15	12–14	2	Yes
16	13–15	3	

SPACE – pause / continue

Screen 18

Activity Number	Activity Name	Elec Fitters Allocated	Critical Activity
3	2–4	5	
4	3–4	4	
6	4–6	2	
8	5–8	2	
10	6–8	5	
12	8–9	2	
15	12–14	2	Yes
16	13–15	3	
18	14–16	1	Yes
20	15–18	2	
22	16–18	3	
24	18–19	4	Yes

12 Activities use this resource, 3 are on a Critical Path

Any key to continue

Screen 19

Activity Number	Activity Name	Con Trades Allocated	Critical Activity

***** Resource not required at any time *****

Any key to continue

Screen 20

When the user is presented with a Summary Screen he will be asked (Screen 21) to select a priority to be levelled.

Summary of Resource Requirements:

Mechanical Fitters:	Needed by 12 activities, 3 are on a Crtical Path
Electrical Fitters:	Needed by 12 activities, 3 are on a Crtical Path
Constructive Trades:	Not Required
Painters:	Not Required
General Labour:	Not Required

Enter priority resource type:

(M) : Mechanical Fitters – MF
(E) : Electrical Fitters – EF
(C) : Constructive Trades – CT
(P) : Painters – PT
(G) : General Labour – GL

Current Data File: **PROJ.DAT**

Screen 21

Screen 22 shows such a selection—the program will automatically save results with the correct identifiable file extension.

Summary of Resource Requirements:

Mechanical Fitters:	Needed by 12 activities, 3 are on a Crtical Path
Electrical Fitters:	Needed by 12 activities, 3 are on a Crtical Path
Constructive Trades:	Not Required
Painters:	Not Required
General Labour:	Not Required

Enter priority resource type:

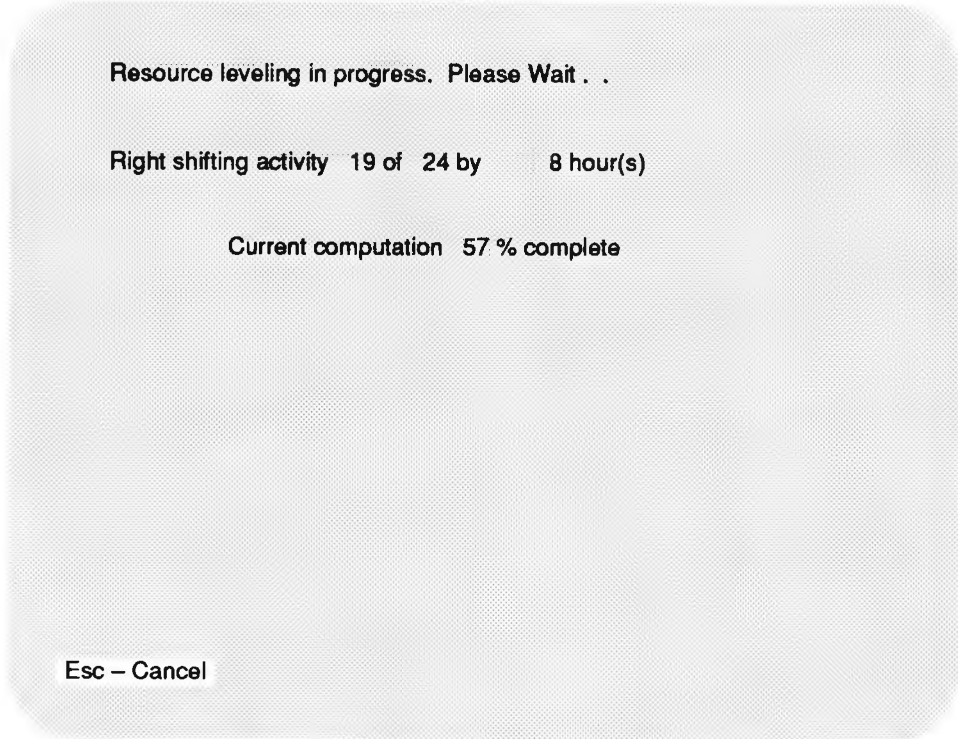
(M) : **Mechanical Fitters** – MF
(E) : Electrical Fitters – EF
(C) : Constructive Trades – CT
(P) : Painters – PT
(G) : General Labour – GL

*** Results will be saved in: PROJ_M.LEV ***

Any key to continue

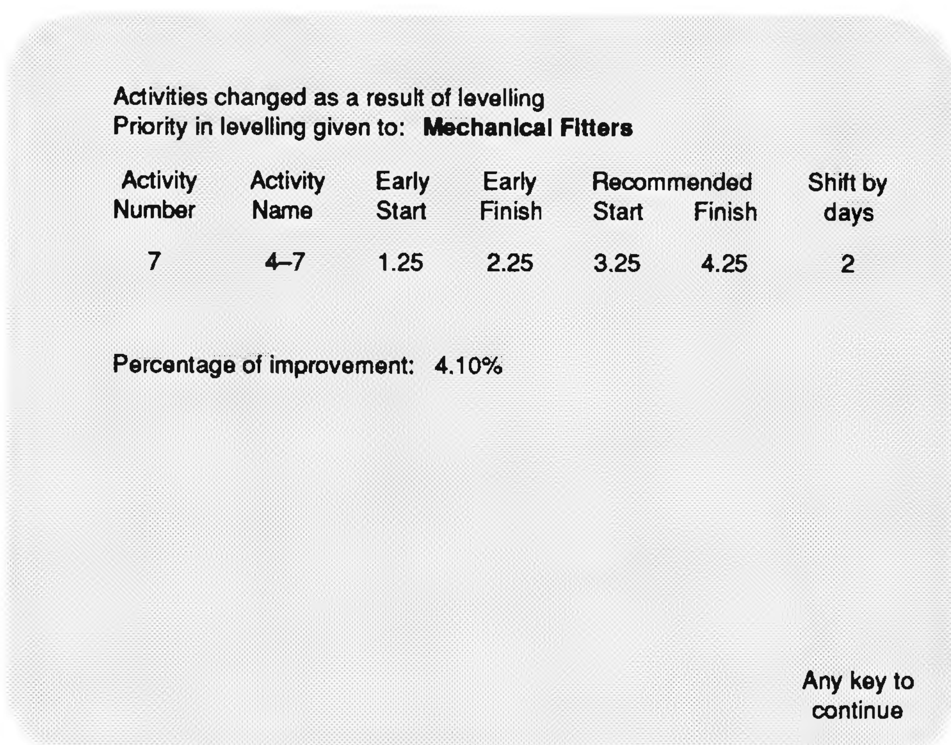
Screen 22

Screen 23 is one presented to a user during computation of resource levelling.



Screen 23

When levelling is completed activities that have changed are shown in Screen 24.



Screen 24

Users, on returning to the PERT PRINT AND LEVELLING MENU can view results of levelling by selecting <F5> and then a file as in Screen 25.

CHRIS_E.LEV CHRIS_M.LEV CHRIS–D_E.LEV PRJ–D_E.LEV

Screen 25

Results are shown as in Screen 26. A user can obtain a Hard Copy by ENABLING the printer from PERT PRINT AND LEVELLING MENU and using <TAB>.

Data File: PERT1B.DAT
Priority in leveling given to: Mechanical Fitters

Activity Number	Activity Name	Early Start	Early Finish	Recommended Start	Recommended Finish	Shift by days
171	140–142	284.73	299.22	306.73	321.22	22
174	145–143	284.82	297.30	285.82	298.30	1
175	117–146	280.43	295.68	287.43	302.68	7
180	131–150	241.17	258.01	273.17	290.01	32
181	130–151	240.98	253.83	245.98	258.83	52
183	143–153	298.20	313.10	321.20	336.10	3
184	147–153	307.07	322.65	320.07	335.65	13
185	153–155	322.65	330.95	335.65	343.95	13
191	150–156	258.01	273.50	298.01	313.50	40
192	156–157	282.65	299.41	313.65	330.41	31
199	157–158	299.41	321.74	330.41	352.74	31

Percentage of improvement: 9.35%

Any key to continue

Screen 26

After selecting and <KEY> user returns to PRINT AND LEVELLING MENU. By choosing <F6> the user is able to modify activities. Screen 27 presents the user Modification Menu.

ACTIVITY MODIFICATION MENU

The resources or times for particular activities can be modified and the PERT Analysis performed on these modified values.

NOTE: Any modifications performed now will be carried over in any subsequent operation unless the original Data File is freshly reloaded at the PERT Main Menu.
An option to save these modifications in an alternative Data File is available when exiting the Print and Levelling Menu.

Available Options:

F1 : Modify Resource Allocations only
F2 : Modify Times only
F3 : Modify all activity details

Esc : Return to Print and Levelling Menu

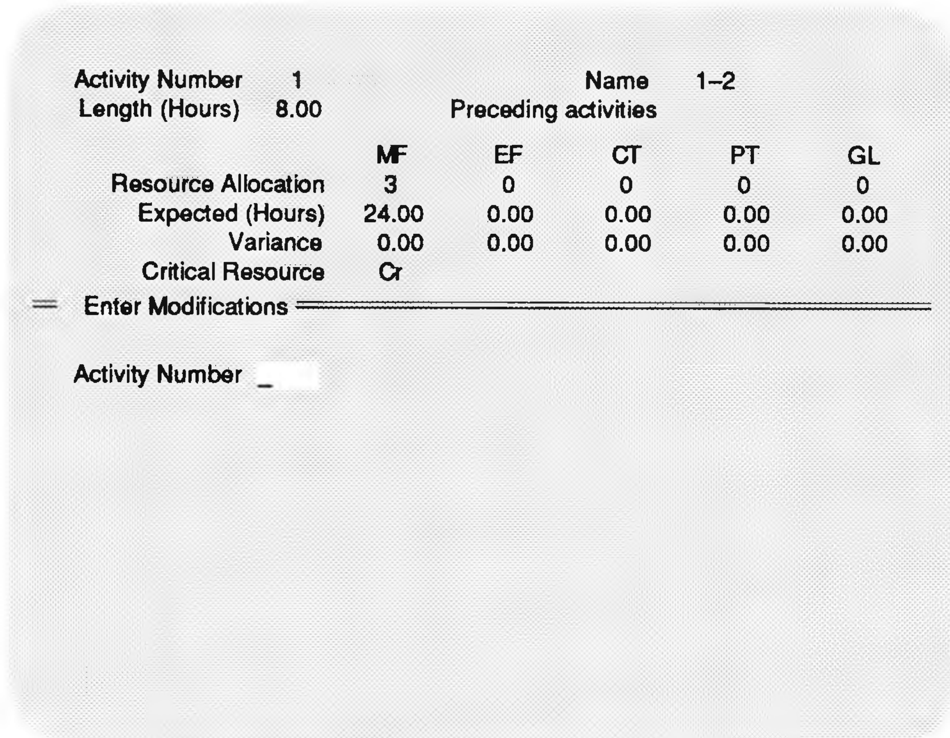
Screen 27

When <F1> is selected the user is asked which activity needs to be changed.

Activity Number : _

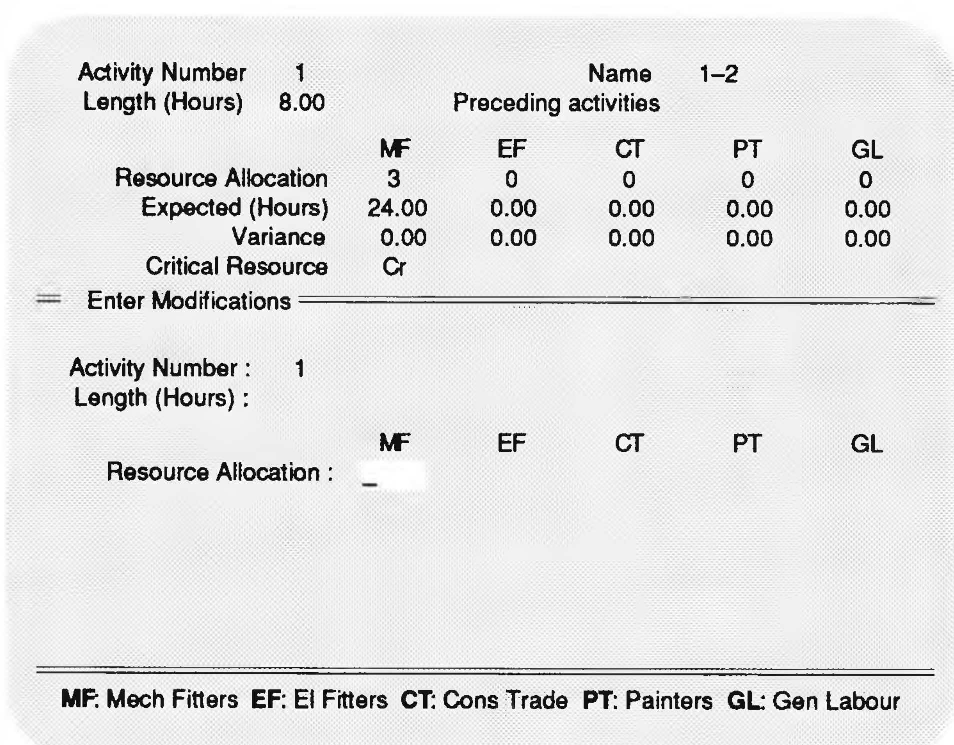
Screen 28

On selecting an activity number and <RETURN>. Users are asked on Screen 29 to confirm activity to be changed.



Screen 30

On confirmation the user can enter changes to Resource Allocation in Screen 30. Enter resource allocation for each trade and <RETURN>. <RETURN> in any resource will enter a zero automatically.



Screen 30

On completion of resource allocations, confirmation as to correctness will be requested as in Screen 31. On positive confirmation the user will be asked to modify another activity. If no further modifications are required <RETURN> will return to PERT MODIFICATION MENU. If <F2>, to modify times only, is selected the user will be asked for activity number.

Activity Number 1 Name 1–2
Length (Hours) 8.00 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation	3	0	0	0	0
Expected (Hours)	24.00	0.00	0.00	0.00	0.00
Variance	0.00	0.00	0.00	0.00	0.00
Critical Resource	Cr				

Enter Modifications

Activity Number : 1
Length (Hours) : 8.00

	MF	EF	CT	PT	GL
Resource Allocation :	3	0	0	0	0
Critical Resource :	Cr				

MF: Mech Fitters **EF**: El Fitters **CT**: Cons Trade **PT**: Painters **GL**: Gen Labour
Is Input correct (Y/N)? _

Screen 31

Activity Number 1 Name 1–2
Length (Hours) 8.00 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation	3	0	0	0	0
Expected (Hours)	24.00	0.00	0.00	0.00	0.00
Variance	0.00	0.00	0.00	0.00	0.00
Critical Resource	Cr				

Enter Modifications

Activity Number 1

Are the Expected Times known (Y/N)?

Screen 32

If expected times are known and confirmation (Y)es in Screen 33, the user will then need to complete Expected Hours per resource. <RETURN> will confirm time or enter zero and move to the next cell. Screen 34 after entering all the changed times, asks user to confirm correctness of input.

```
Activity Number    1                    Name     1-2
Length (Hours)     8.00            Preceding activities

                          MF      EF      CT      PT      GL
   Resource Allocation    3       0       0       0       0
      Expected (Hours)    24.00   0.00    0.00    0.00    0.00
              Variance    0.00    0.00    0.00    0.00    0.00
     Critical Resource    Cr
== Enter Modifications ==========================================

Activity Number :   1
Length (Hours) :

                          MF      EF      CT      PT      GL

   Expected (Hours) :     _

=================================================================
MF: Mech Fitters  EF: El Fitters  CT: Cons Trade  PT: Painters  GL: Gen Labour
```

Screen 33

```
Activity Number    1                    Name     1-2
Length (Hours)     8.00            Preceding activities

                          MF      EF      CT      PT      GL
   Resource Allocation    3       0       0       0       0
      Expected (Hours)    24.00   0.00    0.00    0.00    0.00
              Variance    0.00    0.00    0.00    0.00    0.00
     Critical Resource    Cr
== Enter Modifications ==========================================

Activity Number :   1
Length (Hours) :    8.00

                          MF      EF      CT      PT      GL
                          Cr
   Expected (Hours) :     24.00   0.00    0.00    0.00    0.00

=================================================================
MF: Mech Fitters  EF: El Fitters  CT: Cons Trade  PT: Painters  GL: Gen Labour
Is Input correct (Y/N)? _
```

Screen 34

If Expected Times were not known (as a response to questions concerning knowledge of times) the modification screen will look as in Screen 35.

```
Activity Number     1                          Name      1-2
Length (Hours)    8.00                  Preceding activities

  Resource Allocation      MF       EF       CT       PT       GL
     Expected (Hours)      3        0        0        0        0
             Variance    24.00     0.00     0.00     0.00     0.00
    Critical Resource     0.00     0.00     0.00     0.00     0.00
                           Cr
= Enter Modifications ================================================

Activity Number :   1
Length (Hours) :

                           MF       EF       CT       PT       GL
   Expected (Hours) :
           Variance :

  Optimistic (Hours)
      Likely (Hours) :    _
Pessimistic (Hours)
======================================================================
MF: Mech Fitters  EF: El Fitters  CT: Cons Trade  PT: Painters  GL: Gen Labour
```

Screen 35

Completion of each column will look look Screen 36.

```
Activity Number     1                          Name      1-2
Length (Hours)    8.00                  Preceding activities

                           MF       EF       CT       PT       GL
 Resource Allocation :     3        0        0        0        0
    Expected (Hours) :   24.00     0.00     0.00     0.00     0.00
            Variance :    0.00     0.00     0.00     0.00     0.00
   Critical Resource :     Cr
= Enter Modifications ================================================

Activity Number :   1
Length (Hours) :

                           MF       EF       CT       PT       GL

    Expected (Hours) :   24.00     _
            Variance :    0.00

 Optimistic (Hours) :    24.00
     Likely (Hours) :    24.00
Pessimistic (Hours) :    24.00
______________________________________________________________________
MF: Mech Fitters  EF: El Fitters  CT: Cons Trade  PT: Painters  GL: Gen Labour
```

Screen 36

Completion of all cells will be met by a question confirming correctness of the data.

Activity Number 1 Name 1–2
Length (Hours) 8.00 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation					
Expected (Hours)	3	0	0	0	0
Variance	24.00	0.00	0.00	0.00	0.00
Critical Resource	0.00	0.00	0.00	0.00	0.00
	Cr				

= Enter Modifications =

Activity Number : 1
Length (Hours) : 8.00

	MF	EF	CT	PT	GL
Expected (Hours) :					
Variance :	24.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00
Optimistic (Hours)	Cr				
Likely (Hours) :	24.00	0.00	0.00	0.00	0.00
Pessimistic (Hours)	24.00	0.00	0.00	0.00	0.00
	24.00	0.00	0.00	0.00	0.00

MF: Mech Fitters **EF**: El Fitters **CT**: Cons Trade **PT**: Painters **GL**: Gen Labour
Is Input correct (Y/N)? _

Screen 37

If <F3> had been selected from ACTIVITY MODIFICATION MENU and if Expected Times known will enable the user to change both activity logic and Expected Times as in Screen 38.

Activity Number 1 Name 1–2
Length (Hours) 8.00 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation	3	0	0	0	0
Expected (Hours)	24.00	0.00	0.00	0.00	0.00
Variance	0.00	0.00	0.00	0.00	0.00
Critical Resource	Cr				

= Enter Modifications =

Activity Number : 1 Name _
Length (Hours) : Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation :					
Expected (Hours) :					

MF: Mech Fitters **EF**: El Fitters **CT**: Cons Trade **PT**: Painters **GL**: Gen Labour

Screen 38

Completion of this modification option will determine critical resource for the activity as in Screen 39—confirmation of input will be requested.

Activity Number 1 Name 1–2
Length (Hours) 8.00 Preceding activities

	MF	EF	CT	PT	GL
Resource Allocation	3	0	0	0	0
Expected (Hours)	24.00	0.00	0.00	0.00	0.00
Variance	0.00	0.00	0.00	0.00	0.00
Critical Resource	Cr				

= Enter Modifications

Activity Number : 1 Name 1–2
Length (Hours) : 8.00 Preceding activities 0

	MF	EF	CT	PT	GL
Resource Allocation :	3	0	0	0	0
Expected (Hours) :	24.00	0.00	0.00	0.00	0.00
Critical Resource :	Cr				

MF: Mech Fitters **EF**: El Fitters **CT**: Cons Trade **PT**: Painters **GL**: Gen Labour
Is Input correct (Y/N)?

Screen 39

If Expected Times had not been known the full modification screen will appear as in Screen 40.

Activity Number 1 Name 1–2
Length (Hours) 8.00 Preceding activities

Resource Allocation	MF	EF	CT	PT	GL
Expected (Hours)	3	0	0	0	0
Variance	24.00	0.00	0.00	0.00	0.00
Critical Resource	0.00	0.00	0.00	0.00	0.00
	Cr				

= Enter Modifications

Activity Number : 1 Name 1–2
Length (Hours) : 8.00 Preceding activities 0

	MF	EF	CT	PT	GL
Expected (Hours) :	3	0	0	0	0
Variance :	24.00	0.00	0.00	0.00	0.00
	0.00	0.00	0.00	0.00	0.00
Optimistic (Hours)	Cr				
Likely (Hours) :	24.00	0.00	0.00	0.00	0.00
Pessimistic (Hours)	24.00	0.00	0.00	0.00	0.00
	24.00	0.00	0.00	0.00	0.00

MF: Mech Fitters **EF**: El Fitters **CT**: Cons Trade **PT**: Painters **GL**: Gen Labour
Is Input correct (Y/N)?

Screen 40

Once all modifications have been completed the program will rerun the PERT Analysis again after confirmation of Time Units and activity information shown. User will return to Screen 41 indicating the temporary file name (the file with modifications).

PERT PRINT AND LEVELING MENU

Available options are:

F1 : Print Critical Paths
F2 : Print all Activity Starts, Finishes and Floats
F3 : Produce a Daily Project Work Table
F4 : Level Resources
F5 : Print a Resource Leveling Results File
F6 : Modify an Activity
F7 : Activity Print Menu
TAB : Toggle HardCopy Mode; Currently DISABLED

Esc : Exit Print and Leveling Menu

Current Data File: **$TEMP$.DAT** Original File: **PROJ.DAT**

Screen 41

ACTIVITY PRINT MENU

Available options are:

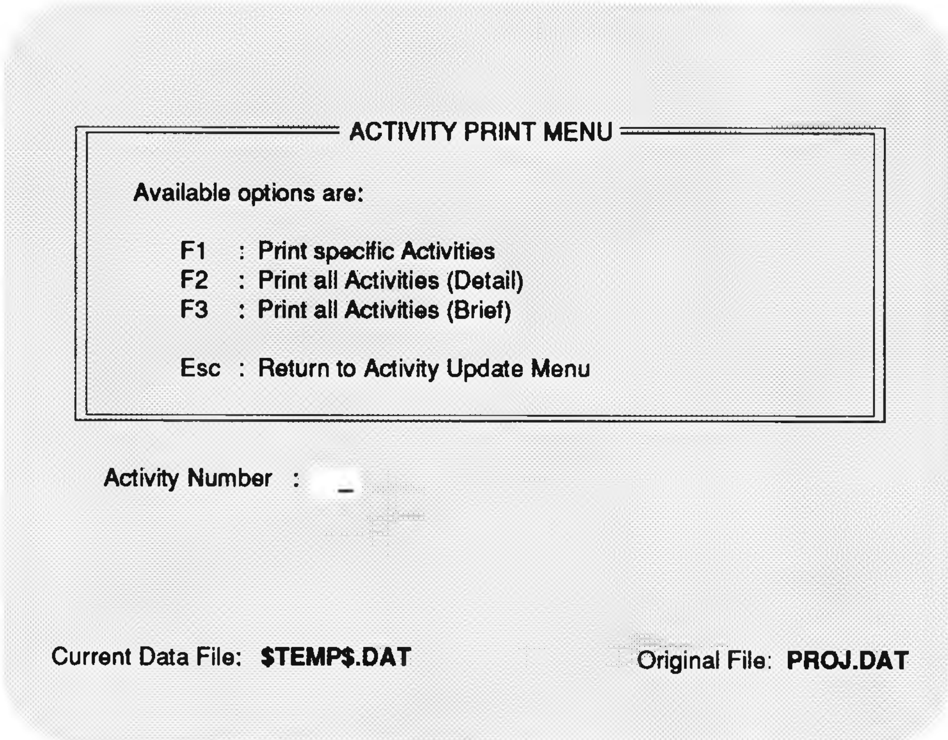
F1 : Print specific Activities
F2 : Print all Activities (Detail)
F3 : Print all Activities (Brief)

Esc : Return to Activity Update Menu

Current Data File: **$TEMP$.DAT** Original File: **PROJ.DAT**

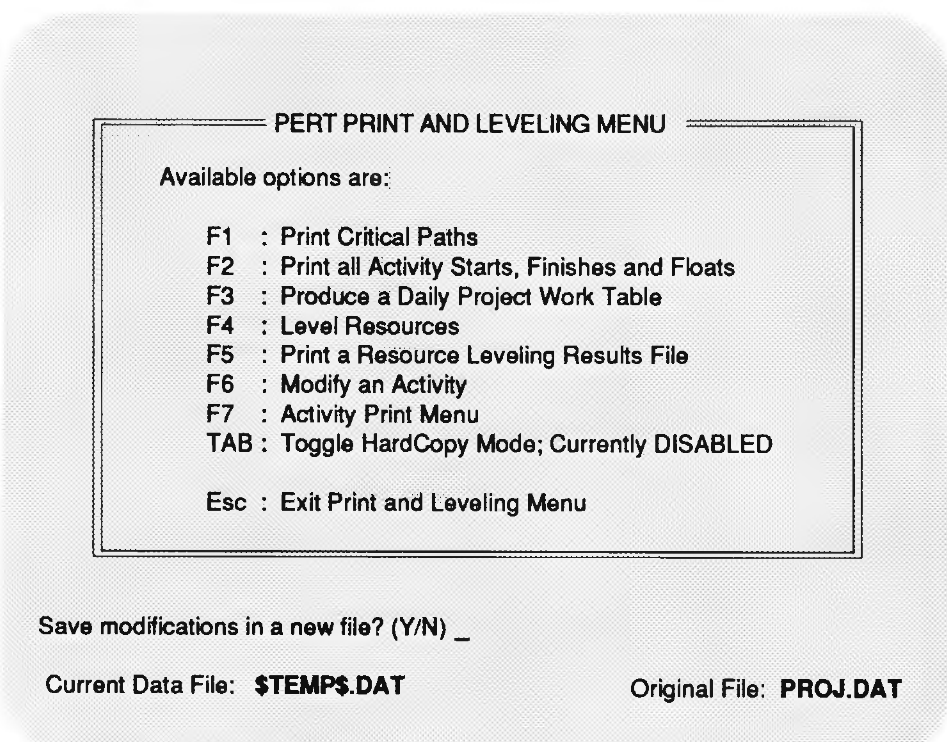
Screen 42

<F7> form the PERT PRINT AND LEVELLING MENU will enable the user to print out various details of activities in the PERT network Screen 43 will appear when selecting <F1>.

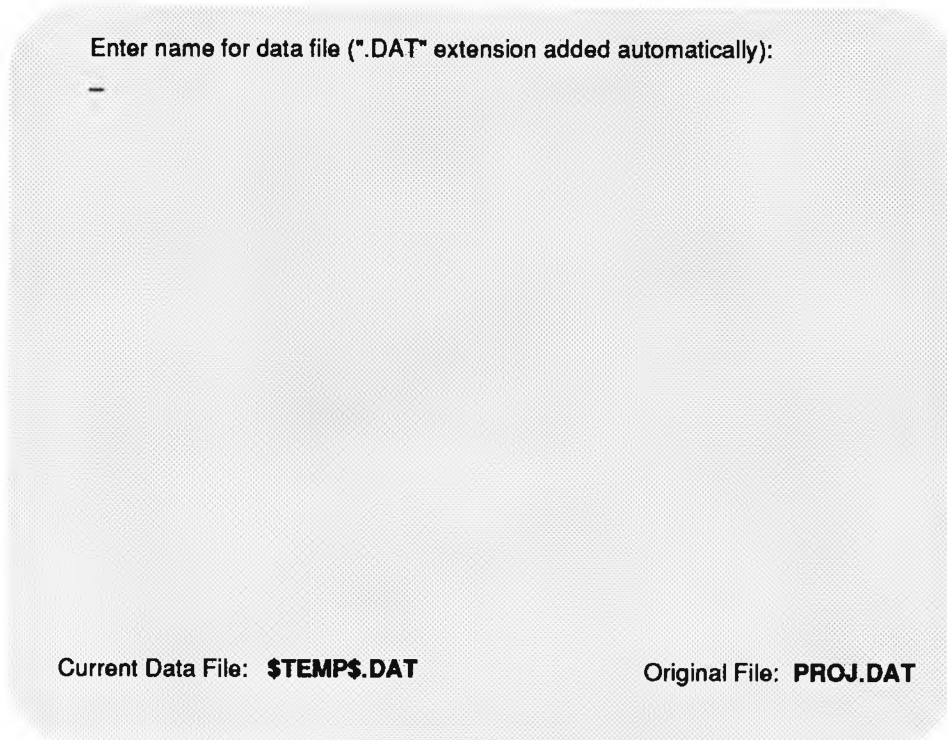


Screen 43

The user, in Screen 44, will be able to save a new Data File (with modifications) under a new name, Screen 45.

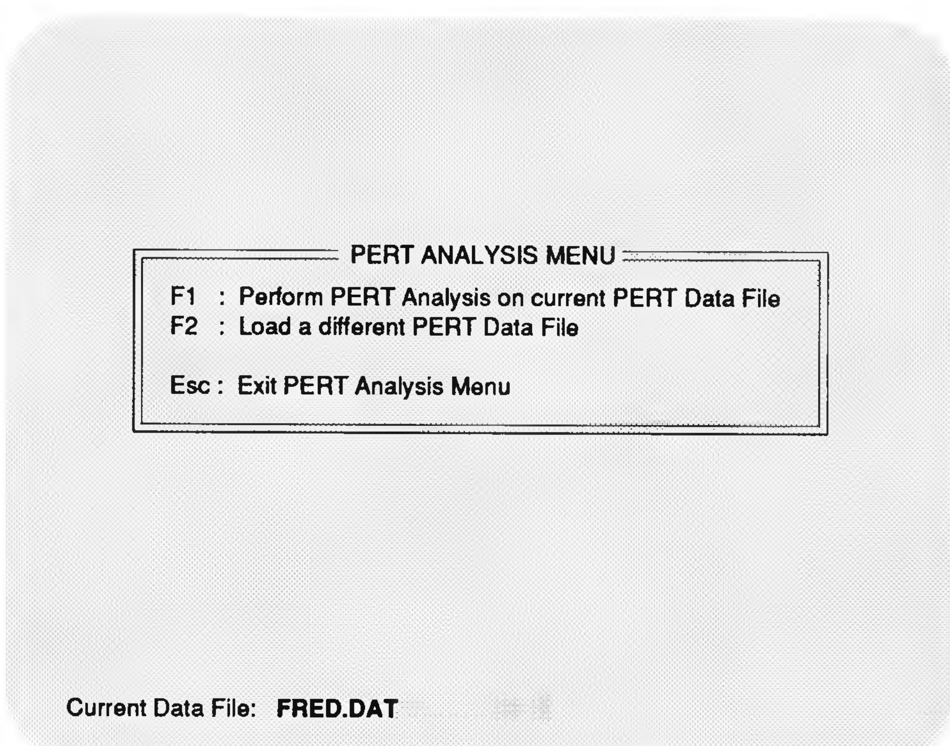


Screen 44



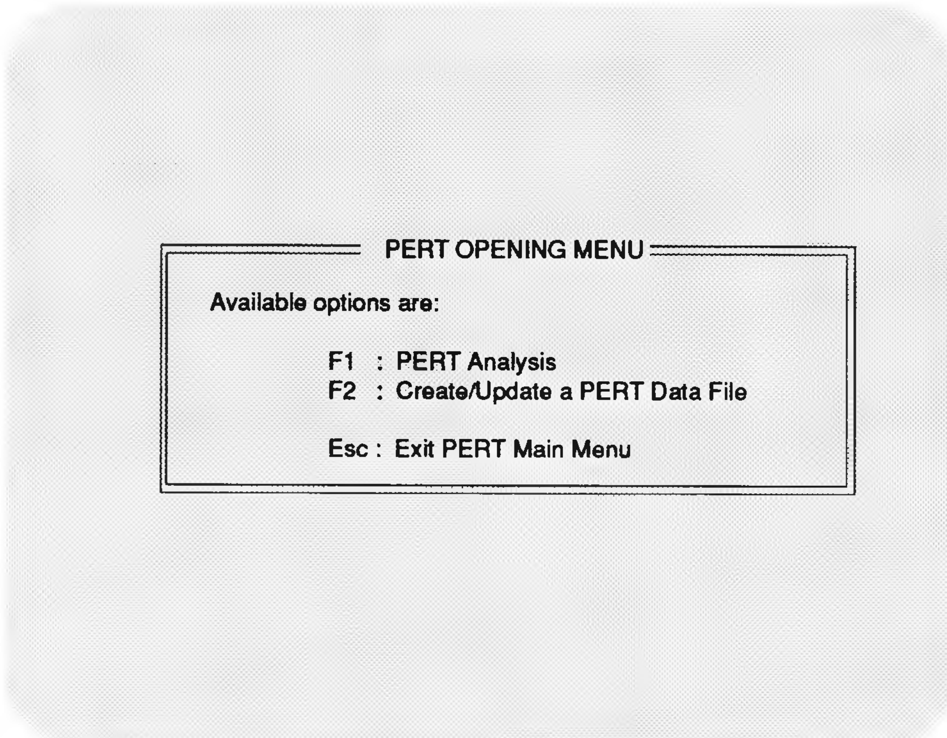
Screen 45

Once file name has been given the user will return to PERT ANALYSIS MENU so either a PERT Analysis can be carried out or new Data File loaded.



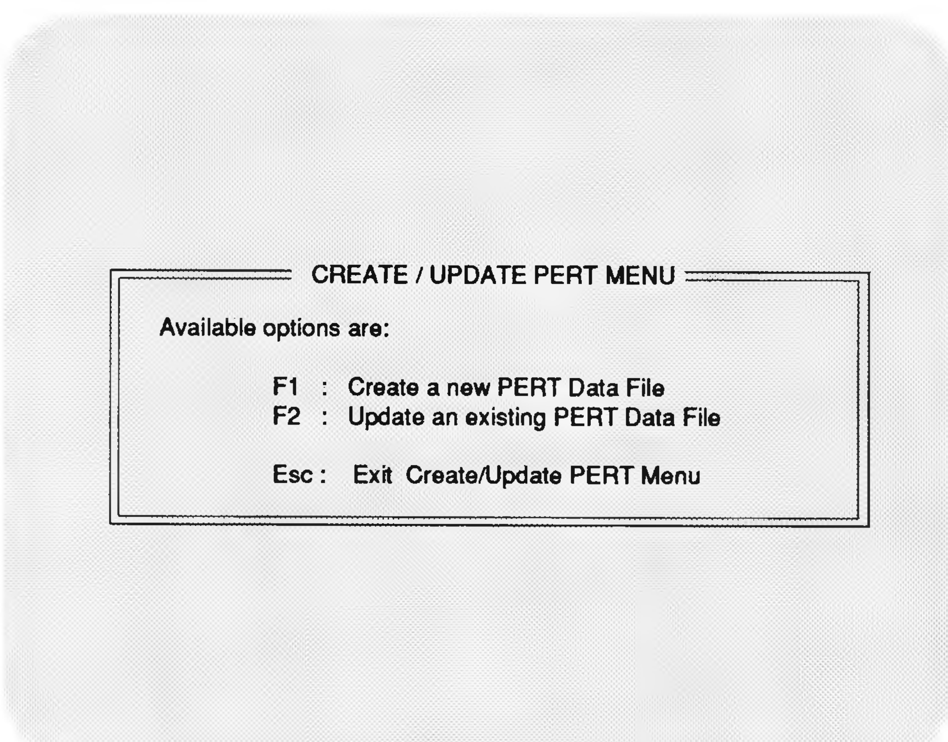
Screen 46

If <ESC> is selected user returns to PERT OPENING MENU



Screen 47

If the user requires to create a new file he selects <F2> from PERT OPENING MENU.



Screen 48

Once new file has been named the user will be asked if expected times are known.

```
Enter name for data file (".DAT" extension added automatically): FRED
Is the expected time for each activity known  (Y/N)? _

                    ============ NOTE ============
                    If times for activities are a mixture of known
                    Expected Times and unknown, type "N" and put in
                    Expected Times as a response to questions for
                    Optimistic, Likely and Pessimistic Times.
```

Screen 49

On a positive response user will commence to insert data/information required to construct a PERT.

```
No  Name  Preceding activities                    Expec  Variance

 1  _
```

Screen 50

Once activity time and preceding activities have been insert the user will be asked for details of Resource Allocations (for each of the five types). <RETURN> provides a zero response.

No	Name	Preceding activities	Expec	Variance
1	1–2	0		

Resource Allocation
Mechanical Fitters: _ Expected Time (Hours):

Screen 51

Once information is entered, confirmation will be requested, Screen 52.

No	Name	Preceding activities	Expec	Variance
1	1–2	0	8.00	0.00

Is input line correct (Y/N)? _

Screen 52

If a positive response is made the user will be asked if any more information is required to be inserted in the new PERT.

No	Name	Preceding activities	Expec	Variance
1	1-2	0	8.00	0.00

More Input? (Y/N) _

Screen 53

If response to question in Screen 49 had been negative the user would be asked for the following information, Screens 54, and 55.

No	Name	Preceding activities	Optim	Likely	Pessim	Expec	Variance
1							

Screen 54

No	Name	Preceding activities	Optim	Likely	Pessim	Expec	Variance
1	1–2	0					

Mech Fitters: _ Optimistic T: Likely T: Pessimistic T:

Screen 55

If there are no more inputs to be made then user returns to CREATE/UPDATE PERT MENU. From this menu <F2> will enable existing file to be updated.

ACTIVITY UPDATE MENU

F1 : Add Activity
F2 : Modify Activity
F3 : Modify Activity Times
F4 : Modify Activity Resources
F5 : Detailed Activity Display
F6 : Summary Display of all Activities
F7 : Activity Print Menu

Esc : Exit Activity Update Menu

Current Data File: **FRED.DAT**

Screen 56

If the user is asks to modify or add an activity that either does not exist or is not the next number in sequence, a warning message, as in Screen 57, will be given. Otherwise, correct add/modify screens will appear.

Activity Number: 11

*** Illegal Activity Number ***

Any key to continue

Screen 57

Once user has created and run PERT (through various options) the simulation option is used to run various LRA scenarios. From PROMISS Main Menu <F2> is selected. Screen 58 provides user with a warning.

The Simulation Option was selected. For this you need to have already run the PERT Network Schedule and produced a Project Resource Table. If this is not the case, press <Esc> now to return to PROMISS Opening Menu and select the F1 option to run the PERT and produce the table. Press <ENTER> now if the table is available.

Please note that a maximum of nine simulations can be carried out in a single session. Should you wish to exceed this limit, exit the Simulation Menu and select it again from the Opening Menu. Old records will not be saved.

Screen 58

Once <RETURN> key has been pressed SIMULATION MAIN MENU will appear. Users can either Load or View a Weekly Work Table. If <F1> is selected Screen 59 will appear.

SIMULATION MAIN MENU

This Option can be used once a Resource Profile has been created by the PERT Weekly Work Table Option (Extension .TAB).

F1 : Load a Weekly Work Table
F2 : View a Weekly Work Table

Esc : Exit the Simulation Main Menu

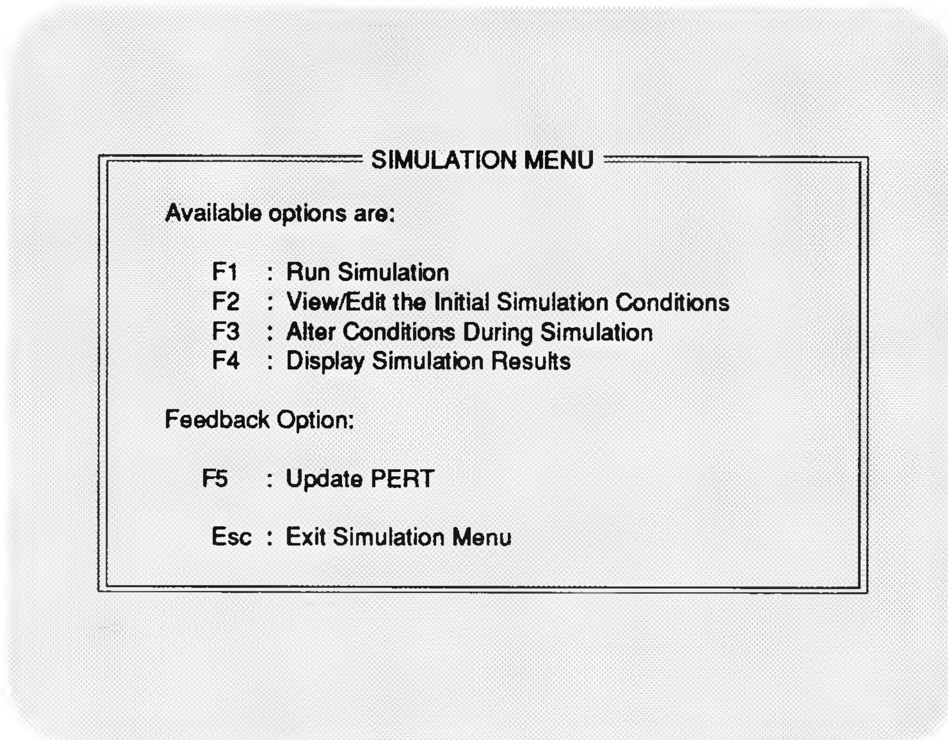
Screen 59

Screen 60 allows, after <F1> has been selected, a user to choose a suitable file.

CHRIS.TAB PERT1B.TAB PROJ2-D.TAB

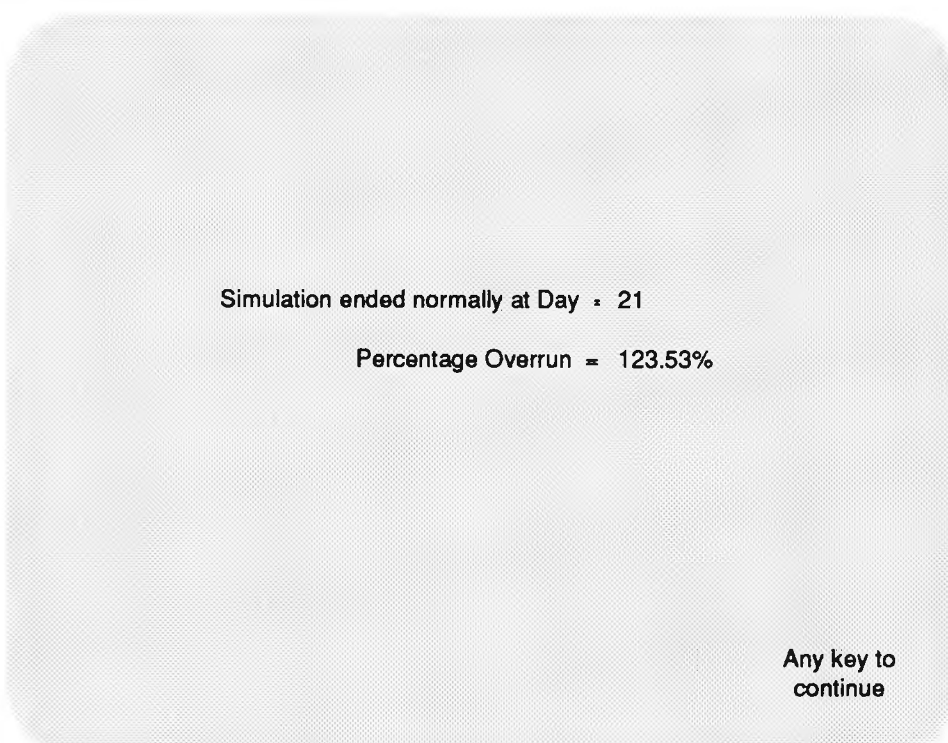
Screen 60

Once a file has been loaded the SIMULATION MENU will appear allowing a user to either run the simulation, view/edit or alter various conditions and display results.



Screen 61

On selecting <F1> the simulation will run. Screen 62 shows the end of simulation result.



Screen 62

Once simulation is complete user can return to MAIN SIMULATION MENU and selecting <F2> Screen 63 will appear.

INITIAL CONDITIONS

		MF	EF	CT	PT	GL	Project
1:	Target Completion Date (W)	4	4	4	4	4	
2:	Average Salary per Week	500	500	500	500	500	
3:	Training Time	8	8	8	8	8	
4:	Available Team	20	20	20	20	20	
5:	Project Team	17	9	0	0	0	
6:	Work Hours per Week	40	40	40	40	40	
7:	Managers Aggressiveness						1.00
8:	Ceiling on Labour Resources	(Dictated by size of Available Team)					NO

Enter the Number of the Condition to Alter OR <Esc> to Accept

MF: Mech Fitters EF: El Fitters CT: Cons Trade PT: Painters GL: Gen Labour

Screen 63

On either accepting or changing initial conditions user can select <F3> from MAIN SIMULATION MENU for the SIMULATION INTERACTION MENU (Screen 64).

SIMULATION INTERACTION MENU

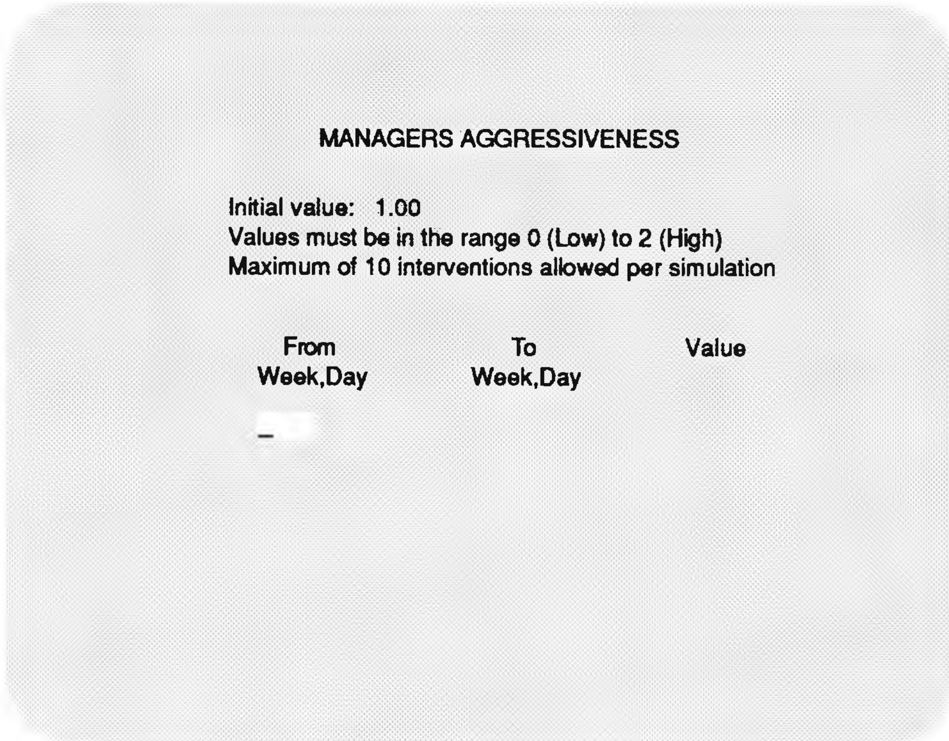
Available options are:

F1 : Alter Managers Aggressiveness
F2 : Alter the Number of Hours Worked per Week
F3 : Alter the Available Team – Mechanical Fitters
F4 : Alter the Available Team – Electrical Fitters
F5 : Alter the Available Team – Constructive Trades
F6 : Alter the Available Team – Painters
F7 : Alter the Available Team – General Labour

Esc : Exit

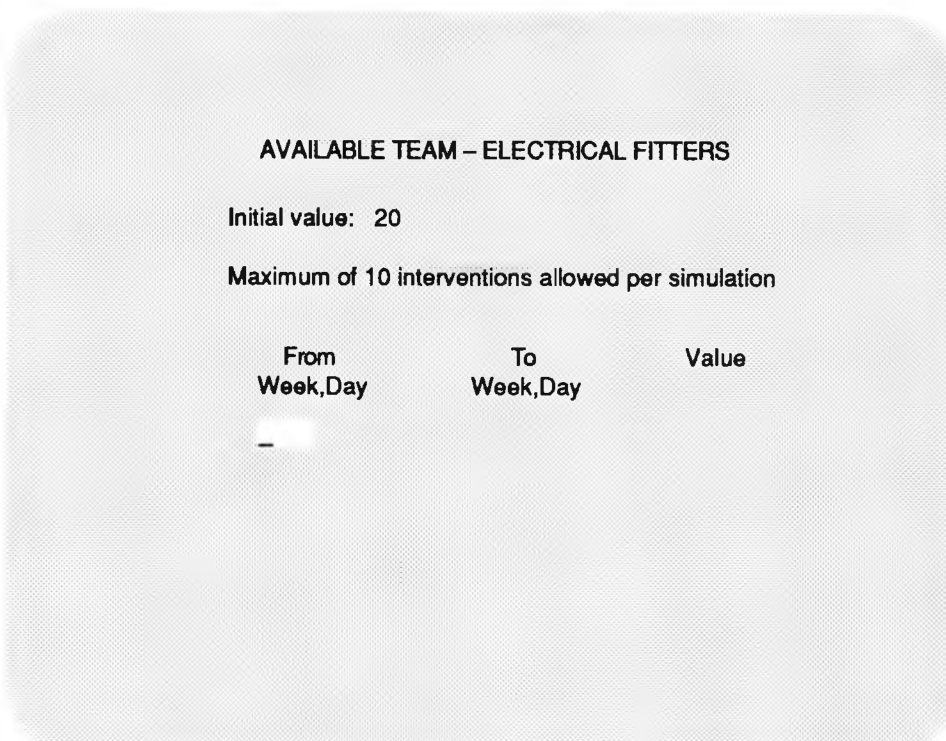
Screen 64

If the user wishes to interact with the model and change for example Managers Aggressiveness <F1> during a simulation, Screen 65 will appear.



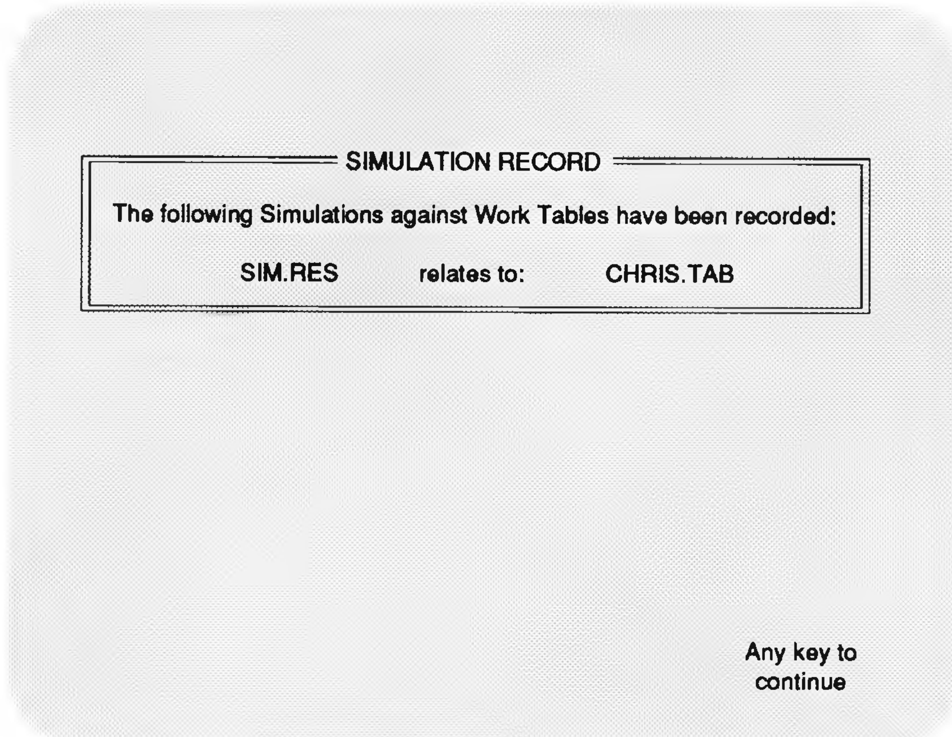
Screen 65

<F4> will allow the user to change, in this example numbers of available Electrical Fitters, see Screen 66.



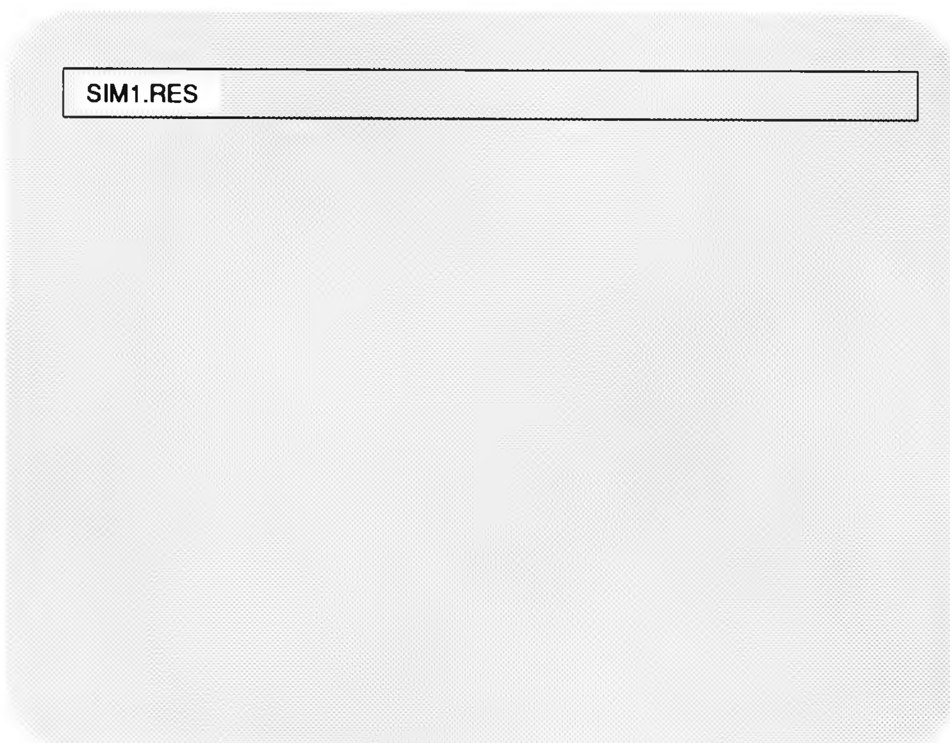
Screen 66

Once changes have been implemented the simulation should be run again. PROMISS keeps a record of all simulations (up to ten per secession). The user will be presented after selecting <F4> from SIMULATION MENU a SIMULATION RECORD as in Screen 67.



Screen 67

User should then select simulation file for results.



Screen 68

The user is able to view and print out results of seventeen variables from Screen 69. Results can be produced in either weekly or daily formats.

SIMULATION VIEWING MENU

Please choose viewing options:

A : Required Team Sizes
B : Desired Team Sizes
C : Project Cost
D : Actual Progress
E : Completed Work With Faults
F : Project Cost Estimate
G : Cumulative Person Weeks
H : Available Team Sizes
I : Reported Fraction Complete
J : Fraction Project Complete
K : Perceived Work Remaining
L : Work to Complete
M : Planned Work
N : Work Completion
O : Total Rework
P : Outstanding Work
Q : Weekly Progress

SPACE : Toggle Time Unit – Currently WEEKS
TAB : Toggle Hard Copy Mode – Currently DISABLED
Esc : Exit Simulation Viewing Menu

Screen 69

From SIMULATION MENU a user can select the feedback option (Screen 70) to update and produce a new PERT with information from a selected simulation run.

FEEDBACK MENU

Available options are:

F1 : Perform Feedback
F2 : View a Feedback Results File

Tab : Toggle Hard Copy Mode – Currently DISABLED

Esc : Exit PERT Main Menu

Screen 69

Once the initial feedback analysis has been completed a daily progress output is displayed as in Screen 70 and 71.

Day	Planned Work	Actual Progress	% Completed —Planned—	% Completed —Actual—
1	32.96	24.72	0.25	0.19
2	32.96	24.59	0.50	0.37
3	32.96	24.54	0.75	0.55
4	32.96	24.53	1.00	0.74
5	32.96	24.57	1.26	0.92
6	34.89	24.66	1.52	1.11
7	34.19	26.48	1.78	1.31
8	33.66	26.22	2.04	1.50
9	36.50	25.25	2.32	1.69

SPACE – pause / continue

Screen 70

Day	Planned Work	Actual Progress	% Completed —Planned—	% Completed —Actual—
401	0.00	65.46	100.00	96.08
402	0.00	50.65	100.00	96.46
403	0.00	41.36	100.00	96.77
404	0.00	46.04	100.00	97.12
405	0.00	43.67	100.00	97.45
406	0.00	44.64	100.00	97.78
407	0.00	32.88	100.00	98.03
408	0.00	38.42	100.00	98.31
409	0.00	18.75	100.00	98.46
410	0.00	28.62	100.00	98.67
411	0.00	23.67	100.00	98.85
412	0.00	26.40	100.00	99.05
413	0.00	25.03	100.00	99.23
414	0.00	25.70	100.00	99.43
415	0.00	25.42	100.00	99.62
416	0.00	25.53	100.00	99.81
417	0.00	25.63	100.00	100.00

Critical Path Length: 362.45 Days
Simulated Project Length: 417.00 Days

Any key to continue

Screen 71

Screen 72 shows the final output from new PERT created from simulation model derived data. Comparison can now be made between "Planned" and "Actual" Activity Length, Start and Finish Times. Critical Activities are also identified.

Feedback File: PERT1B.FBK

Act No	Activity Name	Length (D) Planned	Length (D) Actual	Planned Start	Planned Finish	Actual Start	Actual Finish	
1	1-2	5.45	6.85	0.00	5.45	0.00	6.85	
2	3-4	6.52	8.20	0.00	6.52	0.00	8.20	
3	5-6	8.06	10.08	0.00	8.06	0.00	10.08	
4	7-8	13.27	16.66	0.00	13.27	0.00	16.66	
5	9-10	22.66	27.87	0.00	22.66	0.00	27.87	
6	11-12	5.00	6.27	0.00	5.00	0.00	6.27	
7	13-14	13.79	17.31	0.00	13.79	0.00	17.31	
8	15-16	10.19	12.82	0.00	10.19	0.00	12.82	
9	17-18	16.12	20.14	0.00	16.12	0.00	20.14	Cr
10	2-19	21.76	26.54	5.45	27.21	6.85	33.38	
11	4-20	14.86	18.12	6.52	21.38	8.20	26.32	
12	6-20	8.79	10.84	8.06	16.84	10.08	20.92	
13	8-10	5.32	6.26	13.27	18.59	16.66	22.92	

SPACE – pause / continue Cr : Critical

Screen 72

Once the initial feedback analysis has been performed the Hard Copy Option can be selected from the FEEDBACK MENU.

The user can exit PROMISS by returning back through the menu tree using <ESC> until the DOS prompt appears on the screen.

Appendix 6

Levelling Times for Datum Test Network (PERT1B)

The following resource levelling times where recorded on other various Personal Computers (PCs) running the logic based simulation model (PERT1B)—datum test network. Times are quoted with a maths/numerical processor operating where appropriate.

KEY
RS–Right Hand Shift
LS–Left Hand Shift

Running on PC with a 8086 CPU (Amstrad) and a clock speed of 12Mhz with 8087 numerical coprocessor working.

FILE
PERT1BM.LEV (Priority given to Mechanical Fitters)

Time		Activity	Time units (in units of 8 hours–1 workday)
2215	Started		
0035	RS	27	168
0120		38	392
0210		50	272
0655		101	56
1145		119	296
2050	LS	108	888
2130		109	376
2230		145	248
2305		150	112
2345		157	144
2355		159	456
0020		164	560
0135		181	368
0152		190	312
0218	Finished		

Total Processing Time 28 h 3 min
Improvement in LRA—9.35%

The following levelling exercise provides state of calculations at twenty minute intervals. Same file running on a PC with a 80386 CPU (IBM PS/2—Model 70) and a clock speed of 20Mhz without 80387 numerical coprocessor working.

Time	Activity	Time units (in units of 8 hours–1 workday)
1415 Started		
1435	RS 4	344
1455	11	240
1515	16	24
1535	21	80
1555	28	48
1615	39	12
1635	48	96
1655	52	272
1715	69	440
1735	28	40
1755	80	848
1815	82	640
1835	89	80
1855	97	312
1915	102	296
1935	106	240
1955	108	336
2015	111	56
2035	114	208
2055	115	632
2115	119	136
2135	125	48
2155	140	296
2215	144	120
2235	147	400
2255	152	8
2315	159	96
2335	160	288
2355	164	48
0015	166	40
0035	175	88
0055	191	24
0108 Change from RS → LS		
0115	LS 2	80
0135	6	216
0155	22	208
0215	100	0
0235	125	40
0255	156	128
0315	163	152
0335	165	248
0355	171	0
0415	192	240
0423 Finished		

Total Processing Time 14 h 7 min

Times below are for same data file but with changes in labour priority.

PERT1BE.LEV (Priority given to Electrical Fitters)
Total Processing Time 14 h 42 min
Improvement in LRA—10.78%

PERT1BC.LEV (Priority given to Constructive Trades)
Total Processing Time 12 h 43 min
Improvement in LRA—8.13%

PERT1BP.LEV (Priority given to Painters)
Total Processing Time 11 h 40 min
Improvement in LRA—8.48%

PERT1BG.LEV (Priority given to General Labour)
Total Processing Time 15 h 35 min
Improvement in LRA—8.10%

Various other types of PCs were used to run the data in file PERT1B, with the following results:

8088 CPU (IBM.PC/XT) running with a clock speed of 4.77Mhz and 8087 numerical coprocessor working. Approximate extrapolation after 48hrs running:

9 Days for RS plus 4 for LS

80386 CPU (Research Machines) running at 16Mhz without 80387 numerical coprocessor:

21 h 52 min

80386 CPU (IBM PS/2—Model 80) running at 25Mhz without 80387 numerical coprocessor.

5 h 36 min

80486 CPU (Viglen) running at 25Mhz with integrated numerical coprocessor working:

1 h 56 min

80486 CPU (Viglen) running at 25Mhz without integrated numerical coprocessor working:

5 h 18 min

Appendix 7

Levelling Tables

Below are five tables produced from file PERT1B with each resource levelled in turn. These tables provide the project manager with details of each activity change. Original Early Start and Early Finish times are shown so that a comparison can be made against the Recommended Start and Finish times suggested by the levelling algorithm. The number of days each activity has been shifted is also shown.

Data File: PERT1B.DAT

Priority in levelling given to: Constructive Trades

Activity Number	Activity Name	Early Start	Early Finish	Recommended Start	Recommended Finish	Shift by days
5	9-10	0.00	22.66	46.00	68.66	46
10	2-19	5.45	27.21	21.45	43.21	16
14	10-21	22.66	34.51	62.66	74.51	40
15	12-22	5.00	21.50	11.00	27.50	6
21	22-28	21.50	22.33	68.50	69.33	47
54	48-56	85.31	90.61	93.31	98.61	8
56	48-58	85.31	111.68	94.31	120.68	9
73	64-69	92.04	112.33	114.04	134.33	22
81	73-75	101.16	117.79	156.16	172.79	55
83	63-77	171.72	187.21	184.72	200.21	13
87	57-81	93.41	106.22	112.41	125.22	19
89	81-80	106.22	118.31	111.23	123.31	5
92	77-86	187.21	191.46	190.21	194.46	3
104	72-95	98.66	113.89	101.66	116.89	3
108	72-98	98.66	120.47	231.66	253.47	133
117	94-100	97.32	110.28	101.32	114.28	4
122	106-105	146.81	156.59	186.81	196.59	40
145	121-123	192.39	216.02	211.39	235.02	19
148	125-118	204.39	218.46	258.39	272.46	54
150	124-126	190.11	211.17	191.11	212.17	1
152	122-128	212.32	239.73	214.32	241.73	2
155	129-130	224.62	240.98	271.62	287.98	47
156	129-131	224.62	241.17	264.62	281.17	40
158	126-132	211.17	223.45	274.17	286.45	63
159	98-134	146.73	159.12	268.73	281.12	122
165	137-138	208.39	222.76	281.39	295.76	73
168	139-140	253.78	284.73	275.78	306.73	22
169	138-141	222.76	237.04	295.76	310.04	73
171	140-142	284.73	299.22	306.73	321.22	22
175	117-146	280.43	295.68	293.43	308.68	13
177	146-147	295.68	307.07	308.68	320.07	13
184	147-153	307.07	322.65	315.07	330.65	8
187	152-154	282.65	297.16	302.65	317.16	20
191	150-156	258.01	273.50	260.01	275.50	2
199	157-158	299.41	321.74	330.41	352.74	31

Percentage of improvement: 8.13%

Data File: PERT1B.DAT

Priority in levelling given to: General Labour

Activity Number	Activity Name	Early Start	Early Finish	Recommended Start	Recommended Finish	Shift by days
2	3-4	0.00	6.52	17.00	23.52	17
7	13-14	0.00	13.79	17.00	30.79	17
12	6-20	8.06	16.84	10.06	18.84	2
13	8-10	13.27	18.59	22.27	27.59	9
40	26-44	45.26	49.49	65.26	69.49	20
48	50-51	69.58	80.12	73.58	84.12	4
52	43-55	62.73	73.22	64.73	75.22	2
62	54-61	76.36	83.43	79.36	86.43	3
73	64-69	92.04	112.33	114.04	134.33	22
75	68-70	112.33	123.14	120.33	131.14	8
77	58-83	111.68	131.47	114.68	134.47	3
80	61-74	85.00	92.81	90.00	97.81	5
83	63-77	171.72	187.21	172.72	188.21	1
88	58-82	111.68	120.78	118.68	127.78	7
89	81-80	106.22	118.31	122.22	134.31	16
91	83-85	131.47	152.39	149.47	170.39	18
97	84-88	134.35	151.61	171.35	188.61	37
99	85-90	152.39	172.05	168.39	188.05	16
109	96-98	135.58	146.73	237.58	248.73	102
115	101-103	135.58	156.89	196.58	217.89	61
123	104-107	165.83	186.78	197.83	218.78	32
125	105-108	180.51	194.80	240.51	254.80	60
132	111-113	225.65	254.26	226.65	255.26	1
134	109-110	222.32	239.65	233.32	250.65	11
137	114-117	253.77	280.43	260.77	287.43	7
139	112-118	218.30	238.90	251.30	271.90	33
140	89-119	151.81	175.46	166.81	190.46	15
144	121-122	192.39	212.32	205.39	225.32	13
147	119-125	175.46	204.39	180.46	209.39	5
148	125-118	204.39	218.46	211.39	225.46	7
151	123-127	216.02	232.92	217.02	233.92	1
154	128-127	239.73	249.64	250.73	260.64	11
155	129-130	224.62	240.98	271.62	287.98	47
157	127-132	249.64	266.80	269.64	286.80	20
158	126-132	211.17	223.45	272.17	284.45	61
159	98-134	146.73	159.12	211.73	224.12	65
160	102-134	172.10	185.35	268.10	281.35	96
162	107-136	209.67	235.97	211.67	237.97	2
163	108-137	194.80	208.39	218.80	232.39	24
165	137-138	208.39	222.76	273.39	287.76	65
168	139-140	253.78	284.73	273.78	304.73	20
170	141-142	237.04	248.86	302.04	313.86	65
171	140-142	284.73	299.22	286.73	301.22	2
173	116-145	272.73	284.82	273.73	285.82	1
174	145-143	284.82	297.30	285.82	298.30	1
177	146-147	295.68	307.07	304.68	316.07	9
181	130-151	240.98	253.83	287.98	300.83	47
182	132-152	266.80	282.65	273.80	289.65	7
183	143-153	298.20	313.10	316.20	331.10	18
184	147-153	307.07	322.65	320.07	335.65	13
185	153-155	322.65	330.95	335.65	343.95	13
187	152-154	282.65	297.16	301.65	316.16	19
190	151-156	253.83	266.47	300.83	313.47	47
192	156-157	282.65	299.41	313.65	330.41	31
199	157-158	299.41	321.74	330.41	352.74	31

Percentage of improvement: 8.10%

Data File: PERT1B.DAT

Priority in levelling given to: Mechanical Fitters

Activity Number	Activity Name	Early Start	Early Finish	Recommended Start	Recommended Finish	Shift by days
1	1-2	0.00	5.45	5.00	10.45	5
3	5-6	0.00	8.06	50.00	58.06	50
4	7-8	0.00	13.27	48.00	61.27	48
6	11-12	0.00	5.00	18.00	23.00	18
11	4-20	6.52	21.38	34.52	49.38	28
12	6-20	8.06	16.84	58.06	66.84	50
13	8-10	13.27	18.59	16.27	21.59	3
19	24-26	26.69	28.25	52.69	54.25	26
22	28-29	22.33	25.29	61.33	64.29	39
24	8-30	13.27	18.59	16.27	21.59	3
29	19-36	27.21	39.35	30.21	42.35	3
48	50-51	69.58	80.12	90.58	101.12	21
81	73-75	101.16	117.79	107.16	123.79	6
92	77-86	187.21	191.46	193.21	197.46	6
97	84-88	134.35	151.61	156.35	173.61	22
99	85-90	152.39	172.05	156.39	176.05	4
111	92-100	140.32	163.54	148.32	171.54	8
114	101-102	135.58	172.10	231.58	268.10	96
123	104-107	165.83	186.78	209.83	230.78	44
124	105-107	180.51	209.67	196.51	225.67	16
125	105-108	180.51	194.80	223.51	237.80	43
135	110-116	239.65	261.22	242.65	264.22	3
136	113-116	254.26	272.73	255.26	273.73	1
144	121-122	192.39	212.32	203.39	223.32	11
148	125-118	204.39	218.46	258.39	272.46	54
152	122-128	212.32	239.73	228.32	255.73	16
156	129-131	224.62	241.17	258.62	275.17	34
157	127-132	249.64	266.80	268.64	285.80	19
159	98-134	146.73	159.12	149.73	162.12	3
160	102-134	172.10	185.35	268.10	281.35	96
163	108-137	194.80	208.39	226.80	240.39	32
164	134-138	185.35	199.39	281.35	295.39	96
165	137-138	208.39	222.76	281.39	295.76	73
168	139-140	253.78	284.73	275.78	306.73	22
169	138-141	222.76	237.04	295.76	310.04	73
171	140-142	284.73	299.22	306.73	321.22	22
174	145-143	284.82	297.30	285.82	298.30	1
175	117-146	280.43	295.68	287.43	302.68	7
180	131-150	241.17	258.01	273.17	290.01	32
181	130-151	240.98	253.83	245.98	258.83	5
183	143-153	298.20	313.10	321.20	336.10	23
184	147-153	307.07	322.65	320.07	335.65	13
185	153-155	322.65	330.95	335.65	343.95	13
191	150-156	258.01	273.50	298.01	313.50	40
192	156-157	282.65	299.41	313.65	330.41	31
199	157-158	299.41	321.74	330.41	352.74	31

Percentage of improvement:9.35%

Data File: PERT1B.DAT

Priority in levelling given to: Electrical Fitters

Activity Number	Activity Name	Early Start	Early Finish	Recommended Start	Recommended Finish	Shift by days
1	1-2	0.00	5.45	22.00	27.45	22
4	7-8	0.00	13.27	28.00	41.27	28
8	15-16	0.00	10.19	17.00	27.19	17
21	22-28	21.50	22.33	41.50	42.33	20
22	28-29	22.33	25.29	42.33	45.29	20
28	20-34	21.38	22.26	30.38	31.26	9
30	23-26	27.36	45.26	31.36	49.26	4
56	48-58	85.31	111.68	95.31	121.68	10
62	54-61	76.36	83.43	79.36	86.43	3
67	41-65	61.19	82.56	65.19	86.56	4
82	74-76	92.81	102.02	113.81	123.02	21
89	81-80	106.22	118.31	111.23	123.31	5
91	83-85	131.47	152.39	142.47	163.39	11
96	80-88	176.71	197.43	182.71	203.43	6
105	72-75	98.66	112.99	121.66	135.99	23
108	72-98	98.66	120.47	231.66	253.47	133
109	96-98	135.58	146.73	242.58	253.73	107
114	101-102	135.58	172.10	216.58	253.10	81
119	99-104	149.90	165.83	155.90	171.83	6
122	106-105	146.81	156.59	156.81	166.59	10
123	104-107	165.83	186.78	203.83	224.78	38
124	105-107	180.51	209.67	183.51	212.67	3
137	114-117	253.77	280.43	266.77	293.43	13
143	90-121	172.05	192.39	176.05	196.39	4
145	121-123	192.39	216.02	229.39	253.02	37
147	119-125	175.46	204.39	180.46	209.39	5
148	125-118	204.39	218.46	252.39	266.46	48
150	124-126	190.11	211.17	253.11	274.17	63
151	123-127	216.02	232.92	218.02	234.92	2
155	129-130	224.62	240.98	261.62	277.98	37
157	127-132	249.64	266.80	269.64	286.80	20
158	126-132	211.17	223.45	274.17	286.45	63
159	98-134	146.73	159.12	253.73	266.12	107
164	134-138	185.35	199.39	281.35	295.39	96
168	139-140	253.78	284.73	275.78	306.73	22
169	138-141	222.76	237.04	295.76	310.04	73
170	141-142	237.04	248.86	309.04	320.86	72
171	140-142	284.73	299.22	306.73	321.22	22
173	116-145	272.73	284.82	273.73	285.82	1
181	130-151	240.98	253.83	286.98	299.83	46
183	143-153	298.20	313.10	315.20	330.10	17
184	147-153	307.07	322.65	320.07	335.65	13
185	153-155	322.65	330.95	335.65	343.95	13
190	151-156	253.83	266.47	300.83	313.47	47
192	156-157	282.65	299.41	313.65	330.41	31
199	157-158	299.41	321.74	330.41	352.74	31

Percentage of improvement: 10.78%

Data File: PERT1B.DAT

Priority in levelling given to: Painters

Activity Number	Activity Name	Early Start	Early Finish	Recommended Start	Recommended Finish	Shift by days
1	26-20	8.06	16.84	16.06	24.84	8
19	24-26	26.69	28.25	54.69	56.25	28
23	21-29	34.51	47.57	48.51	61.57	14
40	26-44	45.26	49.49	56.26	60.49	11
47	46-51	66.90	81.37	69.90	84.37	3
62	54-61	76.36	83.43	92.36	99.43	16
88	58-82	111.68	120.78	126.68	135.78	15
99	85-90	152.39	172.05	168.39	188.05	16
100	70-91	123.14	134.69	124.14	135.69	1
107	95-97	113.89	130.21	114.89	131.21	1
119	99-104	149.90	165.83	193.90	209.83	44
122	106-105	146.81	156.59	158.81	168.59	12
134	109-110	222.32	239.65	224.32	241.65	2
136	113-116	254.26	272.73	255.26	273.73	1
137	114-117	253.77	280.43	258.77	285.43	5
139	112-118	218.30	238.90	233.30	253.90	15
140	89-119	151.81	175.46	152.81	176.46	1
141	89-120	151.81	166.74	156.81	171.74	5
147	119-125	175.46	204.39	177.46	206.39	2
148	125-118	204.39	218.46	211.39	225.46	7
154	128-127	239.73	249.64	258.73	268.64	19
155	129-130	224.62	240.98	271.62	287.98	47
157	127-132	249.64	266.80	258.64	275.80	9
163	108-137	194.80	208.39	227.80	241.39	33
164	134-138	185.35	199.39	281.35	295.39	96
165	137-138	208.39	222.76	211.39	225.76	3
168	139-140	253.78	284.73	275.78	306.73	22
169	138-141	222.76	237.04	295.76	310.04	73
170	141-142	237.04	248.86	310.04	321.86	73
171	140-142	284.73	299.22	306.73	321.22	22
174	145-143	284.82	297.30	285.82	298.30	1
175	117-146	280.43	295.68	284.43	299.68	4
177	146-147	295.68	307.07	308.68	320.07	13
181	130-151	240.98	253.83	258.98	271.83	18
183	143-153	298.20	313.10	321.20	336.10	23
187	152-154	282.65	297.16	302.65	317.16	20
191	150-156	258.01	273.50	260.01	275.50	2
192	156-157	282.65	299.41	313.65	330.41	31
199	157-158	299.41	321.74	330.41	352.74	31

Percentage of improvement: 8.48%

Index

F

G

H

I

K

L

M

S

T

U

V

W