Methodologies for the Structured Development and Documentation of Manufacturing Planning and Control Systems

Mark Duncan Stirling

Business and Informatics Research Institute

Information Systems Institute

University of Salford, Salford, UK

Submitted in Partial Fulfilment of the Requirements of the

Degree of Doctor of Philosophy, November 2000

If you think you can, you can. And if you think you can't, you're right

Mary Kay Ash

To my Father,

for always believing that I can.

Table of Contents

Table of Contents .... **ii**

Table of Figures .... **vii**

Acknowledgements .... **ix**

Declaration .... **x**

Publications by the Author .... **xi**

About the Author .... **xii**

Abbreviations .... **xiii**

Abstract .... **xiv**

Chapter 1. Introduction to the Project **1**

1.1 Introduction 1

1.2 Background to the Total Technology PhD Scheme 2

1.3 Background to Ferodo Ltd., Chapel-en-le-Frith Site 4

1.4 The MRPII Software Package used – MFG/PRO 8

1.5 Statement of the Area of Concern 9

1.6 Objectives of the Research 10

1.7 Research Framework 11

1.8 Research Methodology 13

1.9 Organisation of the Remainder of the Study 17

Chapter 2. Manufacturing Planning and Control Systems **19**

2.1 Introduction 19

2.2 Early Manufacturing Planning and Control Systems 20

2.3 Manufacturing Resource Planning, (MRPII) 24

2.4 Problems with MRPII Systems 39

2.5 Capacity Planning 44

2.5.1 Master Production Scheduling 45

2.5.2 Rough Cut Capacity Planning 48

2.5.3 Capacity Requirements Planning 49

2.5.4 Finite Capacity Scheduling 50

2.6 Just in Time 53

2.7 Optimised Production Technology 62

2.8 Development of Information Technology 69

2.9 Computer Systems Development 69

2.10 Systems Analysis and Design Methodologies 71
2.11 Modelling Techniques...80
2.12 Database Development 85
2.13 The MRPII Integration Problem 91
2.13.1 Centralised Database Strategy..92
2.13.2 Distributed Database Strategy..93
2.13.3 Hybrid Database Strategy 95
2.14 The Core and Sub-Systems Approach to Systems Development.........................96
2.15 MPC System Current Developments...97
2.15.1 The Extended Enterprise..97
2.15.2 Enterprise Resource Planning...98
2.15.3 Internet in Manufacturing 99
2.15.4 Object-Oriented Programming 104
2.16 The Documentation Problem 107
2.16.1 The International Organisation for Standardisation Procedures...................109
2.16.2 Development of Documentation...113
2.16.2.1 The Minimalist Approach to Documentation 115
2.16.2.2 The Elaboration Approach to Documentation....................................116
2.16.3 Problems with Paper Documentation 118
2.17 The Electronic Approach to System Documentation 120
2.17.1 Document Management Systems 125
2.17.1.1 Document Management Systems..125
2.17.1.2 Groupware Systems...127
2.17.2 Documentation Intranets...129
2.17.3 Problems with the Electronic Approach to System Documentation..........133
2.17.4 Specific Problems with Document Management Systems 134
2.17.5 Specific Problems with Intranets 135
2.18 The Ferodo OE Manufacturing Planning and Control System............................136
Chapter 3. Structured MPC System Development...138
3.1 Introduction 138
3.2 Case Study Companies 140
3.2.1 Ferodo Aftermarket Manufacturing Division..141
3.2.2 Bentley Harris..142
3.2.3 Eurobrakes Ltd..143

3.2.4 Eurofriction Ltd..144
3.2.5 Ferodo Railways Division 145
3.2.6 Case Study Findings 146
3.3 Early Systems Development at Ferodo..148
3.4 Satellite Systems Integrated at Ferodo 149
3.4.1 Quality Analysis System...151
3.4.2 Property Recording System 154
3.4.3 Manufacturing Instructions System..158
3.4.4 Batch Tracking Report 163
3.5 Development of the Systems Development Methodology...............................165
3.5.1 The Three Integration Techniques..169
3.5.1.1 Incorporation...169
3.5.1.2 Common Characteristics of Satellite Systems 171
3.5.1.3 Integrated Satellite Systems 175
3.5.1.4 Joined Satellite Systems 176
3.5.2 The Systems Development Life Cycle Methodology...............................177
3.5.2.1 Change in Business Requirements 180
3.5.2.2 Process Changes...181
3.5.2.3 New Module..183
3.5.2.4 Incorporated Integration 184
3.5.2.5 Satellite System Integration...185
3.5.2.6 Core System Modification...186
3.5.2.7 Independent Package 188
3.5.3 Systems Development Life Cycle Methodology Decision Support...........189
3.5.3.1 Decision Support Questions...189
3.5.3.2 Indicative Ratios 192
3.6 General Applicability of the Methodology..196
3.7 Problems with the Satellite Systems Approach..197
3.8 Structured MPC System Documentation 198
3.9 The Ferodo MPC System Documentation..199
3.10 The Ferodo Intranet..201
3.10.1 The Ferodo Intranet for MRPII Documentation 209
3.11 Further Development of the Systems Development Methodology 210
3.11.1 The MPC System Documentation Template..212

3.11.2 The Four Levels of Documentation 216
3.11.3 The New Systems Development Life Cycle Methodology 222
3.11.3.1 Define System Boundaries...224
3.11.3.2 Modify the Template to Fit the System...227
3.11.3.3 Functionally Decompose Template 227
3.11.3.4 Apply PIPS to Decomposed Template..228
3.11.4 Document Storage and Distribution 229
3.12 General Applicability of the New Methodology...230
Chapter 4. Validation of the Systems Development Life Cycle Methodology 231
4.1 Introduction 231
4.2 Interrogative Validation Techniques 232
4.3 The Validation Interviews...236
4.4 The Validation Interview Results 237
4.5 The Validation Interview Findings 242
4.6 Application of the Systems Development Life Cycle Methodology.................244
4.6.1 Case Study Company A 246
4.6.2 Case Study Company B 248
4.6.3 Case Study Company C 251
Chapter 5. Summary and Conclusions 257
5.1 Summary..257
5.1.1 The Structured Development of MPC Systems 257
5.1.2 The Structured Documentation of MPC Systems 262
5.1.3 Key Elements of the Research..266
5.2 Conclusions..268
5.3 Recommendations for Further Work..271
5.3.1 The Structured Development of MPC Systems 271
5.3.2 The Structured Documentation of MPC Systems 272
Chapter 6. Appendices 273
6.1 Quality Analysis Files 273
6.1.1 Main Datafile..273
6.1.2 Headings Control File 274
6.2 Statistical Process Control, (SPC) 275
6.3 Property Recording Files 278
6.3.1 SPC Rules Control File..278

6.3.2 Standard Tests Control File 279
6.3.3 Item Tests Control File 280
6.3.4 Main Data Storage Datafile...281
6.3.5 Data Entry and Reporting 283
6.4 Manufacturing Instructions Files...288
6.4.1 Index Headings Control File 288
6.4.2 Data Field Headings Control File 289
6.4.3 Main Data Storage Datafile...290
6.4.4 History Datafile 291
6.4.5 Reporting 292
6.5 The Structured Interview Template 294
References 303

Table of Figures

Figure 1: Ferodo Divisional Structure 5
Figure 2: Original Ferodo OE Division Information System 6
Figure 3: New Ferodo OE Division Information System 7
Figure 4: The Action Research Cycle 15
Figure 5: Structure and Flow of Thesis 18
Figure 6: Re-Order Point Theory 21
Figure 7: Effect of Increased Demand on Re-Order Point Theory 22
Figure 8: MRP Logic 24
Figure 9: MRP Calculation for a Disk Brake Pad Product Structure 26
Figure 10: Sample Disc Brake Pad Bill of Materials 26
Figure 11: Closed-Loop MRP 29
Figure 12: MRPII 30
Figure 13: The MRPII Database Structure 33
Figure 14: Master Production Scheduling 46
Figure 15: Table of Machine Capacity against Load 47
Figure 16: Hierarchical Factory Organisation Structure 51
Figure 17: The Deming Cycle 54
Figure 18: Kanban Control Techniques 57
Figure 19: Effects of Lowered Inventory 58
Figure 20: The Input - Output Diagram 80
Figure 21: The IDEF0 Diagram 81
Figure 22: IDEF0 Functional Decomposition 82
Figure 23: Integrated Database Approaches 87
Figure 24: Framework for Planning, Implementing and Managing an Intranet .. 132
Figure 25: Developed Ferodo OE Information System 150
Figure 26: New Systems Database Construction 165
Figure 27: New Systems Internal Construction 166
Figure 28: Summarised Systems Development Life Cycle 177
Figure 29: The Systems Development Life Cycle 178
Figure 30: Techniques Available to Systems Developers 179
Figure 31: The Decision Support Indicative Ratios 192
Figure 32: Top Level MPC System Documentation Template 214
Figure 33: The New Summarised Systems Development Life Cycle 222

Figure 34: New Summarised Systems Development Life Cycle With Feedback...223
Figure 35: The New Systems Development Life Cycle...223
Figure 36: Analysis of Systems Analysis / MRPII Experience...238
Figure 37: Analysis of the Requirement for a Methodology...238
Figure 38: Analysis of the Systems Development Life Cycle Methodology...239
Figure 39: Analysis of the Structured Questions...239
Figure 40: Analysis of the Indicative Ratios...240
Figure 41: Analysis of the Methodology Extension to include Documentation...240
Figure 42: Analysis of the Top Level MPC Systems Documentation Template...241
Figure 43: Analysis of the PIPS Levels of Documentation...241
Figure 44: Results of Validation Interviews...242
Figure 45: Decision Support Questions applied to Case Study Co. A...247
Figure 46: Decision Support Questions applied to Case Study Co. B – Part 1...249
Figure 47: Decision Support Questions applied to Case Study Co. B – Part 2...250
Figure 48: Decision Support Questions applied to Mobile Data – Part 1...252
Figure 49: Decision Support Questions applied to Mobile Data – Part 2...255
Figure 50: Manufacturing Instructions Main Datafile Structure...291

Acknowledgements

Due to the nature of industrial research I am indebted to a number of people, especially:

Dr David Petty, (Federal Mogul Senior Lecturer in Manufacturing) and **Professor Trevor Wood-Harper**, (Professor of Information Systems), my Academic Supervisors, for their help and guidance.

Mr Jeff Bailey, (OE Business Analyst, Ferodo Ltd., my Industrial Supervisor), for all the time he has devoted to answering my questions, for listening to and constructively criticising my developing theories and for supporting and assisting my work at Ferodo.

Mr Henri Ramczyk, (Director Light Vehicle Operations, Ferodo Ltd.), for providing the support necessary for my project and for giving me unrestricted access to Ferodo.

In addition to these individuals special thanks must also go to **John Roberts** and **David Venables** of Ferodo MIS, for fielding many questions and for helping me develop the systems shown in this thesis and to **Dr Mike Sheridan**, of the Ferodo Quality Department, for allowing me to experiment with his quality systems.

Thanks must also be given to the rest of the staff of the Ferodo MIS department for their help with the technical development of the systems described within this thesis.

Finally, thanks are due to my parents for their continued support throughout the project and to my wife Blanka for putting up with the long hours that I have kept over many years without complaining.

Declaration

No portion of this work has been submitted to this, or any other, university or any other institution of learning in support of an application for any other degree or qualification.

Publications by the Author

1. **Stirling, M.D.**, Petty, D.J. and Bailey, J. (1997), *The Implementation of Integrated Business Control Systems in a Manufacturing Company*, Total Technology 20 Year Celebration Poster Session, 26 June.

2. **Stirling, M.D.**, Petty, D.J., Venables, D.J. and Harrison, D.K. (1997), *Development of an Integrated Manufacturing and Quality Control Information System - A Case Study*, Proceedings of the 32nd International Matador Conference, Formerly the International Machine Tool Design and Research Conference, UMIST, 10-11 July, Pages 125-130.

3. Petty, D.J., Sutton, A.D., Bailey, J., Saxby, T.M. and **Stirling, M.D.** (1997), *Manufacturing Planning and Execution Systems to Cope with Batch and Semi-Process Material Flow*, Advances in Manufacturing Technology XI, Proceedings of the 13th National Conference on Manufacturing, Glasgow Caledonian University, 9-11 September, Pages 416-421.

4. Petty, D.J., **Stirling, M.D.**, Travis, L.C. and Bennett, R. (1998), *Finite Capacity Scheduling and Manufacturing Resource Planning Hybrid Production Control Systems: Conditions for Successful Implementation*, Advances in Manufacturing Technology XII, Proceedings of the 14th National Conference on Manufacturing, University of Derby, 7-9 September.

5. **Stirling, M.D.** and Petty, D.J. (2000), *Support for Service Functions: Integration into an ERP Framework*, Supply Chain Practice, Journal of the Supply Chain Knowledge Centre, Cranfield School of Management, Vol. 2, No. 3, Pages 48-55.

6. Petty, D.J., **Stirling, M.D.**, Travis, L.C. and Bennett, R., *Finite capacity / MRPII hybrid control systems: conditions for successful implementation*, Accepted for Publication in: The Journal of Engineering Manufacture (Short Communications): Proceedings of the Institution of Mechanical Engineers, Part B.

7. **Stirling, M.D.**, *Service Functions and ERP Packages – Mutually Exclusive?*, Accepted for Publication in: IT Solutions: Supplement to Manufacturing Management.

8. **Stirling, M.D.**, Petty, D.J. and Travis, L.C., *The Development of Information Systems in Manufacturing Companies*, Submitted to: The International Journal of Operations and Production Management.

About the Author

The author left school in 1991 and started an undergraduate training scheme with AVRO International Aerospace. This training scheme was designed to compliment an engineering degree as part of a thick sandwich course. In 1992 sponsorship was provided by AVRO International Aerospace to undertake a Mechanical Engineering degree at Salford University. The author graduated with first class honours in 1995.

After graduating, a place was awarded on the Total Technology PhD scheme, co-ordinated by UMIST, to do a collaborative PhD with Ferodo Ltd., Chapel-en-le-Frith and Salford University. The project was based on-site at Ferodo Ltd.

The placement with Ferodo involved fulfilling the role of systems analyst, giving close operational support to the Business Analyst / Systems Manager, whilst also pursuing the research objectives of the PhD. In particular, responsibility was assumed for assisting with safeguarding the operation of the Manufacturing Resource Planning package used within Ferodo. Consultancy work was also undertaken for some of the companies within the same group as Ferodo and for suppliers to Ferodo in order to gain a breadth of understanding of business problems.

The author left Ferodo in July 1998 and started working for Minerva Industrial Systems Plc, a leading supply chain management software supplier, as an ERP consultant and then as project manager for the inventory management practice.

Whilst working with Ferodo the author achieved election to the grade of associate member of the Institution of Mechanical Engineers and full, chartered, membership was achieved whilst working for Minerva Industrial Systems plc.

Abbreviations

ASCII	American Standard Code for Information Interchange
BPR	Business Process Re-Engineering
CATWOE	Checkland's System Definition Acronym
DMS	Database Management System
E-R	Entity Relationship
EMail	Electronic Mail
EOQ	Economic Order Quantity
ERP	Enterprise Resource Planning
FCS	Finite Capacity Schedule
GRASP	Graduate Support Programme
HTML	Hypertext Mark-up Language
IDEF	International Definition
INC	Internal Non-Conformance
ISO	International Organisation for Standardisation
ISO9000	International Organisation for Standardisation Procedure 9000
IT	Information Technology
JIT	Just in Time
MPS	Master Production Schedule
MRP	Material Requirements Planning
MRPII	Manufacturing Resource Planning
OE	Original Equipment Division
OPT	Optimised Production Technology
Purchase Order ID	Purchase Order Identification Number
RAM	Random Access Memory
RCCP	Rough Cut Capacity Plan
ROCE	Return on Capital Employed
ROP	Re-Order Point
SPC	Statistical Process Control
SSADM	Structured Systems Analysis and Design Method
WIP	Work in Process
WM	Workflow Management
Work Order ID	Work Order Identification Number
XML	Extensible Mark-up Language

Abstract

Computerised Manufacturing Planning and Control, (MPC), systems are used by many manufacturing organisations and there has been a significant amount of research into the implementation and use of these systems. It is apparent that these systems, once implemented, require continuous development to meet changing business requirements. What is not well understood is the optimal approach for this development process. This thesis presents the findings of a collaborative industrial research project that addresses this issue. The collaborative partner was Ferodo Ltd., Chapel-en-le-Frith, a leading automotive friction product manufacturer. The project was conducted under the Total Technology scheme.

A review of the development of MPC systems is presented. This review considers three approaches to MPC; Manufacturing Resource Planning, (MRPII), Just in Time, (JIT) and Optimised Production Technology, (OPT). It is shown that whilst there is diversity between these approaches and their application in industry, there is convergence between their data structure requirements. The work presented in this thesis is based around the MRPII package used within Ferodo. The research concentrated on defining methodologies for structured systems development, with two main themes:-

1. The development of a multi-stage methodology to assist in the appropriate choice of systems development technique for creation of an effective manufacturing database.

2. Following on from the above, the thesis identifies the need for structured, hierarchical documentation to accompany a manufacturing database. A methodology for creation of this documentation is presented which is based on a pre-defined, top level, template. The methodology uses modelling techniques and defines four levels of documentation to help system developers derive comprehensive documentation from this template. Intranet technology is proposed as a mechanism for providing general access to this documentation.

Proposals for further work are presented which include additional testing of the systems development methodology and creation of templates for different industrial scenarios.

Chapter 1. Introduction to the Project

1.1 Introduction

This chapter introduces the thesis by detailing the background to the research project. It will start by explaining why the project was commissioned and will describe the Total Technology PhD scheme, (section 1.2). The chapter will continue by introducing the collaborating company and the software used, (sections 1.3 and 1.4). It will then set the scene for the research, by showing the origins and objectives of the project, (sections 1.5 and 1.6). The chapter will continue by introducing the research framework, will detail the research methodology adopted, (sections 1.7 and 1.8) and will end by explaining the organisation of the remainder of the project, (section 1.9).

The PhD project was conducted with a collaborating company, Ferodo Ltd., based predominantly on-site at their plant in Derbyshire. Following the implementation of a Manufacturing Planning and Control, (MPC), system in 1994 management were aware that keeping up to date with business practices required structured systems development. To help address this requirement members of the original implementation team were asked to work on continuous development, in particular by being involved in monthly business improvement committee meetings.

Senior management at Ferodo also entered into a three year partnership with the University of Salford to support a Total Technology PhD Researcher investigating how a company might improve their MPC systems once implemented. This was to be done by developing the existing systems within Ferodo in line with business requirements. The development work was to form the basis of research into the use and development of MPC systems conducted by the PhD Researcher at the University of Salford.

1.2 Background to the Total Technology PhD Scheme

In the 1970's it was recognised that many researchers with a desire to achieve a PhD also wanted a senior professional position within industry. Unfortunately industry was finding that a typical PhD graduate had been streamlined towards research and not towards senior professional positions. A more rounded individual, with experience of day to day problems and issues found in the workplace was required, (Wood and Leonard 1978). It was shown that these researchers needed two things from their particular PhD Course:

- Firstly, they needed a practical, broad, industry-based subject to study. This concept is not unusual as many research topics are closely allied to industry;
- Secondly, they needed work experience so as not to find that the three years in academia only helped their employment opportunities within that environment.

As a result of industry voicing these concerns the Total Technology scheme was introduced at UMIST in 1977. This was in keeping with the views of Professor Sir Hugh Ford who originally defined the term '*Total Technology*', (Conway 1993).

Depending upon the research topic a Total Technology researcher can spend varying amounts of time working on-site at a senior level with a company. For example, some purely laboratory-based research may take place largely at a University and factory-based research will be biased towards working within a company.

Research projects are backed up by attendance at appropriate supporting lectures and seminars including the UMIST Graduate Support Programme, 'GRASP'.

The research is always designed to benefit the company and at the same time, to generate original results. Thus the research must not only meet the traditional PhD requirements for originality but also must be sufficiently broad based to encompass a range of areas, for example:

- Sales;
- Marketing;
- Production;
- Logistics;
- Finance.

Total Technology is in essence the study of a total problem rather than of a single focused one. The successful researcher graduates with a PhD but has also been exposed to the discipline of working at a senior level in industry. The Total Technology programme, therefore, is specifically designed to enable successful researchers to achieve a senior industrial position at an early stage of their career.

Despite the work being done to promote the Total Technology scheme there is still general misunderstanding of the differences between the scheme and traditional doctoral research. This is an on-going problem that is being addressed by the introduction of other collaborative Doctoral schemes including the four year Engineering Doctorate approach, (Conway 1993).

1.3 Background to Ferodo Ltd., Chapel-en-le-Frith Site

Ferodo Ltd., Chapel-en-le-Frith, recently part of the Federal Mogul Corporation, was part of the T&N Plc group of companies, which in 1996 had a published turnover of £1956 Million, (T&N Annual Report and Accounts 1996). The company was split into 7 product groups: Bearings; Sealing Products; Composites and Camshafts; Construction Materials and Engineering; Discontinued Operations; Friction Products and Piston Products. Ferodo Ltd. was part of the T&N Friction Products Group, (it is now part of the Federal Mogul General Products Group). The T&N Friction Products Group, in 1996, had a turnover of £329 Million.

The Chapel-en-le-Frith site, (from now on referred to as Ferodo), is responsible for the manufacture of Disc Brake Pads and Drum Brake Linings for both the Original Equipment, (OE) and Aftermarket sectors throughout Europe. The Friction Products Group is responsible for the formulation of the mix that constitutes the friction material used in the manufacture of the Disc Brake Pads and the Drum Brake Linings. The vehicles supplied range from Motorcycles, through Domestic Cars, Commercial Vehicles and Trains.

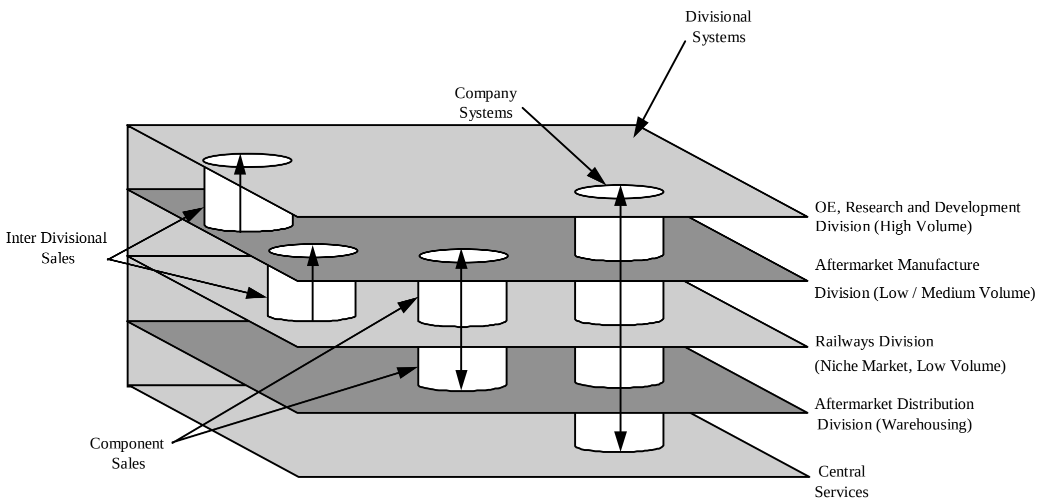
In 1994 Ferodo was broken down into five autonomous product divisions, split on market lines, to cope with this product mix, (Petty and Harrison 1995):

- OE, (Research and Manufacture);
- Aftermarket Manufacture;
- Railways;
- Aftermarket Distribution;
- Central Services.

The OE division is a high volume large batch size manufacturer. The Aftermarket division is a medium volume small batch size manufacturer. The Railways division is a low volume small batch size manufacturer. Whilst these divisions are autonomous they are a single legal entity. The Aftermarket distribution division does not manufacture, it buys from the other divisions and runs a large warehousing operation. Unlike the other divisions it reports independently. Central Services is not a trading entity but provides support to the other divisions.

This is represented in Figure 1 below, (Harrison *et al* 1995), with the links between the divisions illustrated. Note that the company operates by inter-divisional trading, for example the OE division sells product to the Aftermarket Manufacture division.

Figure 1: Ferodo Divisional Structure

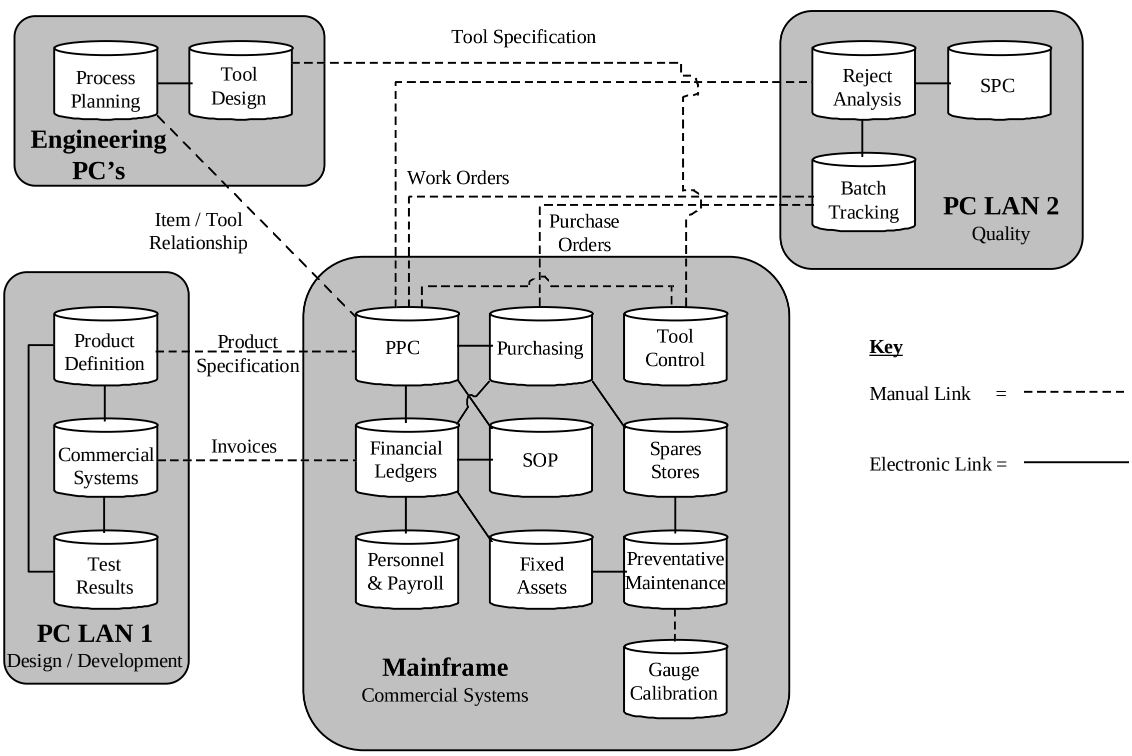


In 1994 Ferodo decided to implement an MPC system and chose a Manufacturing Resource Planning, (MRPII), package, which is in keeping with the actions of many manufacturing companies, (Denton and Hodgson 1997). One result of choosing the divisional structure illustrated in Figure 1 was that each division could independently implement their own systems.

By doing this Ferodo acted as the pilot site for the Friction Products Group. The package chosen for the implementation was MFG/PRO supplied by Minerva Industrial Systems plc, (see section 1.4). MFG/PRO is being implemented across the Friction Products Group. The companies in the Friction Products Group contain a mix of the above divisions. As a result of developing divisional MRPII models first in Ferodo these implementations have been made easier as the models can be copied.

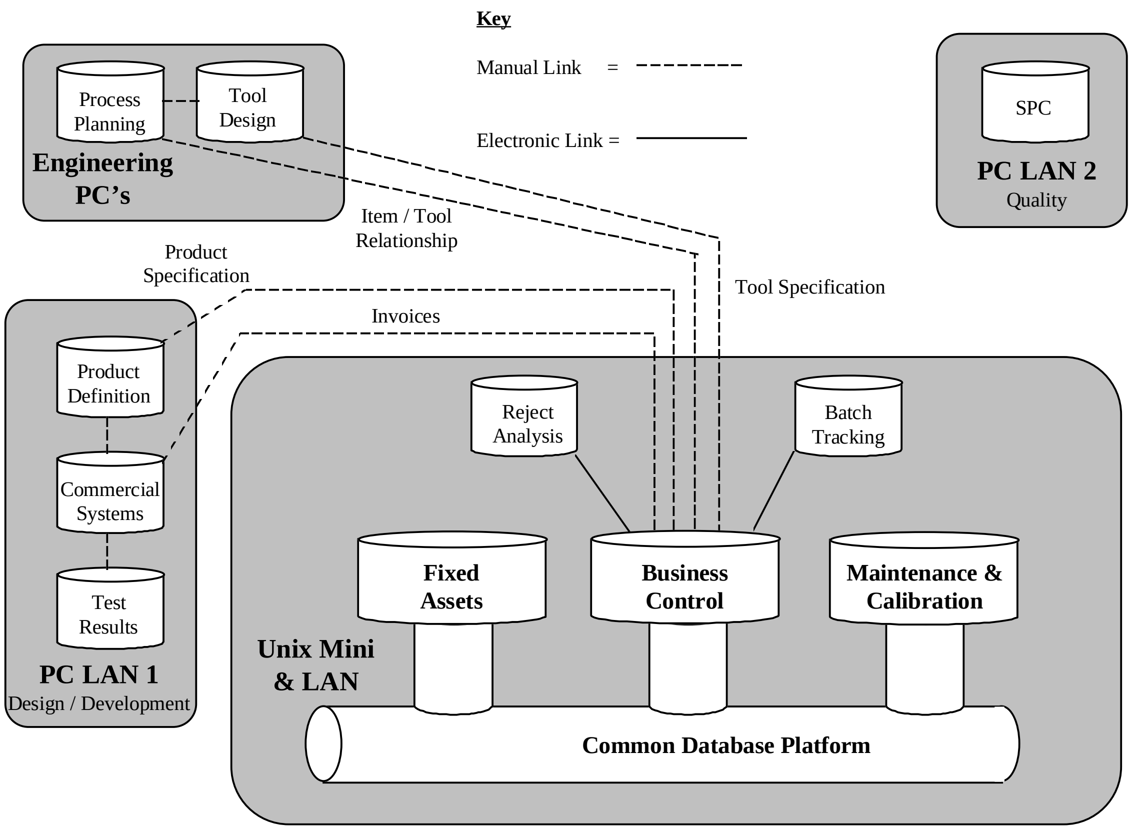
Within the OE Division the implementation of MFG/PRO resulted in a comprehensive restructuring of the original information systems. The original system is illustrated in Figure 2 below, (Harrison *et al* 1995). It can be seen that the system was split into a variety of individual, standalone systems, linked manually, with a few electronic links between the key systems. These systems were designed and implemented separately, leading to poor integration and therefore, duplicated effort, (Harrison *et al* 1995).

Figure 2: Original Ferodo OE Division Information System



As a result of implementing MFG/PRO the information systems were rationalised considerably. The result was to have a core system serving a number of sub-systems. The sub-systems were all selected as they were relatively independent and as such they did not need to be linked to the core system. The structure of the new system is reflected in Figure 3 below, adapted from Harrison *et al*, (1995).

Figure 3: New Ferodo OE Division Information System



This system offers more control for the business allowing direct links between many of the Business Control systems. Whilst this approach yields significantly better performance, the process of systems integration was not undertaken formally or systematically, (Harrison *et al* 1995). The concept and use of sub-systems was not clearly defined in the published material at the time.

1.4 The MRPII Software Package used – MFG/PRO

In keeping with many other MPC Packages, for example SAP and J.D.Edwards, MFG/PRO is an integrated MRPII system that consists of a number of fully interactive modules to assist with control of many parts of the business.

Reading the software brochures and Web-Site information, (Minerva Industrial Systems Web-Site, SAP Web-Site[1]), these packages have many impressive features, including:

- Supply chain management;
- Shop floor data capture;
- Sales, purchasing and distribution order processing;
- Financial management;
- Integrated manufacturing.

These packages support a range of company types, (Wortman *et al* 1996). For example Minerva has case-studies for a number of installations: Courtaulds Chemicals; Kitchen Range Foods; Varity Perkins and Hubbard Group Services, (Minerva Industrial Systems Web-Site).

The systems are based on large relational databases, written using modern fourth generation languages. For example MFG/PRO was developed using a language called Progress. A fourth generation language can be defined as a modern programming language that appears similar to human language, (Webopedia 2000).

---

[1] Due to the increasing use of the Internet it has become increasingly common for authors to use the Internet as a publishing medium. The author accepts that work published on the Internet is not necessarily refereed before publication and that references to documents, so called Internet references, may not always be available, (unlike journals in a library). It has become evident to the author, however, that some Internet references are necessary for the flow of this thesis and as a result some are cited. Every attempt has been made to reduce the frequency of Internet references.

1.5 Statement of the Area of Concern

Significant effort has been focussed on the implementation of Manufacturing Planning and Control systems. This has helped systems implementers determine the best way to go about an implementation – for example the ‘Proven Path’ methodology for implementing MRPII systems first proposed in the 1970’s, (the Proven Path implementation methodology is detailed in section 2.3).

Established theory does not, however, fully explain how to develop an existing implementation. Chapter 2 will show that there has been work done on how to use Linear Goal Programming to determine which of a number of potential development projects should be done so as to maximise the potential return whilst minimising the risk. Chapter 2 will also show case studies of development activities. The development options available to a systems development team have not been so well defined, neither has the best practice to be adopted when selecting an option. As a result of identifying the potential gains to be made in this area, the post implementation development of MPC systems was defined at the start of the PhD research as the area of concern.

PhD research is necessarily a fluid endeavour. During the course of the work defined in Chapter 3 the author determined that the newly developed systems needed to be properly documented and investigated literature about the documentation of MPC systems. The results of this investigation are presented in Chapter 2. It was noticed that the literature available did not specifically relate to the documentation of MPC systems and as a result this was also adopted as an area of concern. This second area of concern is an extension of the original area of concern and as such is covered by the aims and objectives developed to investigate the original area of concern, (see section 1.6).

1.6 Objectives of the Research

Due to the research being industrially based the research objectives must combine an element of contribution to the collaborating company as well as working towards satisfactory research output. To this end the original aims of the project were formally defined as below:

- To review the existing information systems within Ferodo;
- To review the nature of change of these information systems;
- To devise a strategy for development of a company wide information system;
- To implement the strategy within the limits of time available;
- To therefore devise a methodology for the development of business systems.

The hypothesis that was formed was as follows:

- It is possible to define a methodology for the structured, post implementation, development of MPC systems.

In order to achieve the aims and prove or disprove the hypothesis the following detailed objectives were set:

- A detailed understanding of MPC systems was to be gained, from literature review and case-study visits;
- A detailed understanding of the Ferodo MPC systems was to be gained;
- The Ferodo MPC systems were to be developed to increase their effectivity;
- The understanding of MPC systems development gained during the Ferodo development was to be used to devise a methodology for MPC systems development;
- The methodology was to be tested using appropriate techniques in order to help verify that the methodology had utility.

1.7 Research Framework

There are two types of research framework that could be applied to a research project, (Saunders *et al* 2000):

- Theoretical Framework;
- Conceptual Framework.

A theoretical framework is usually determined at the start of a research project and consists of a researcher utilising existing theory when conducting the research. For this approach to be successful the theory must exist in advance, allowing the researcher to test and expand the existing theory. The main drawback of this approach is that establishing a theoretical framework prior to conducting research could potentially put boundaries on the research. The researcher may have difficulty conducting research if the participants in a research project have views that deviate excessively from the theoretical boundaries. In addition the boundaries may themselves bring the research to an artificial end as they may limit the expansion of the research, (Saunders *et al* 2000).

As an alternative a researcher may decide not to establish a theoretical framework at the start of a research project but to start to conduct research and collect data. The researcher can then analyse the data collected and determine how the research should proceed based upon the findings. This approach to research, establishing a conceptual framework, has the following characteristics, (Saunders *et al* 2000):

- The approach is inductive in nature;
- Data collection determines the research path;
- The research does not start with a theoretical framework;
- A clear research purpose needs to be identified to steer the research.

The research objectives detailed in section 1.6 required the researcher to be an active participant in a systems development activity and to formulate the research output as a result of this interaction. As a result of this requirement a conceptual framework approach was adopted.

As explained in section 1.5 the research was to be undertaken into the post implementation development of MPC systems. Authors such as Thomas Wallace, (Wallace 1990) and Oliver Wight, (Wight 1984), demonstrated how to successfully implement these systems. Harrison *et al*, (1995), present a case study of a post implementation development project. Reynolds et al, (1997), present a methodology, using Linear Goal Programming, that helps a systems developer decide which development projects would offer the greatest return with the lowest risk. Following on from the work presented by these authors the aim of this research was to formalise the systems development process. The work was to be conducted in a manufacturing company using an MRPII system and the research findings would be restricted to this subset of company. The research was conducted as below.

Using knowledge gained during the literature review presented in Chapter 2 the author conducted a number of systems development projects within the case study company. Once these projects were completed the author was able to reflect upon these projects and develop a methodology for systems development.

The work on systems development led in turn to identification of the need for structured documentation. This was investigated and a development project undertaken. As a result of the development project the systems development methodology was enhanced.

1.8 Research Methodology

There are three main approaches to research, (Saunders *et al* 2000):

- Experimentation, (*physical or theoretical interaction with a subject*);
- Survey, (*soliciting opinion regarding a subject*);
- Case Study, (*reviewing the work of others in real-life situations*).

The experimentation approach can be further broken down as follows:

- Conceptual Study, (*theoretical analysis of a subject*);
- Mathematical Modelling, (*attempt to prove or disprove a theory mathematically*);
- Laboratory Experiment, (*controlled experimentation in a laboratory environment*);
- Field Experiment, (*experimentation in a real-life environment*);
- Action Research, (*participation in a real-life situation, reflecting on the findings*).

The author was registered on a Total Technology PhD which was to be based on-site within a collaborating company. The research required the author to gain an understanding of the approaches that could be used to develop an MPC system and determine a methodology to structure and control the application of these approaches. For this reason only the following research methodologies could be applied:

- Survey;
- Case Study;
- Field Experiment;
- Action Research.

The research objectives required the author to review the results of development projects within the collaborating company and, therefore, surveys could not be used.

Field Experimentation requires the researcher to repeat experiments to verify results. Industrial projects cannot be repeated and as a result this methodology was also rejected.

As the project was based in an Industrial environment and addressed issues relevant to industry, it was clear that the Action Research methodology should be considered.

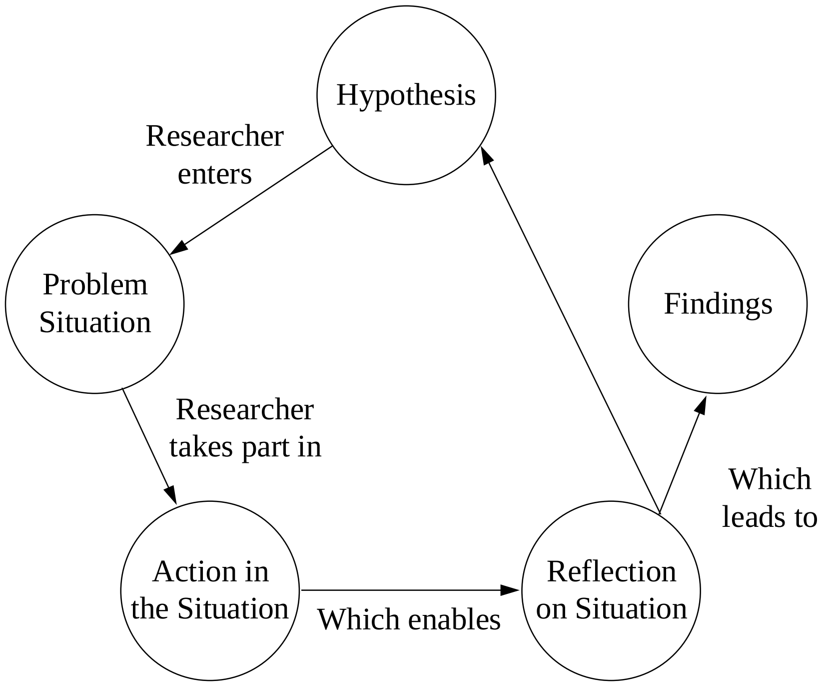
Action Research has been around since the 1950's. One of the earlier authors to define the concept of Action Research was Blum, (1955). Blum described Action Research as having two stages, a diagnostic stage where a situation is reviewed in order to develop theories and a practice stage where the theories are put to the test.

Work on Action Research has been developing at the University of Lancaster, from where there have been a number of published works. Several of the better known works on Action Research were published by Professor Peter Checkland, (Checkland 1981, 1991, Checkland and Scholes 1990, Checkland and Holwell 1998), though there are other authors writing in the field, (Baskerville and Wood-Harper 1996, Wilson 1990).

The Action Research cycle proposed by Blum has been developed since the 1970's to include up to five stages, (Baskerville and Wood-Harper 1996). The new stages allow for results to be developed from testing the theory initially developed and the theory either proved or developed into a new theory. This new theory then re-enters the Action Research cycle if the situation permits.

The Action Research Cycle is shown in Figure 4 below, adapted from Checkland and Holwell, (1998).

Figure 4: The Action Research Cycle



The Action Researcher, therefore, interacts with a company, creating and testing theories by undertaking projects in the company, (Baskerville and Wood-Harper 1996).

Baskerville and Wood-Harper, (1996), present a number of criticisms that have been levelled at the Action Research methodology:

1. It is difficult for researchers to remain impartial when researching within a company;
2. There can be a lack of scientific discipline when working with companies;
3. Action Research projects occur in unique situations, which are usually unrepeatable.

An alternative approach to consider is Case Study Research, (Eisenhartd 1989). In this approach the researcher distances themselves from the problem and develops their theories by monitoring the work of others.

This approach addresses the main criticism of the Action Research methodology, (difficulty in remaining impartial) but does have drawbacks itself. One criticism is also the strength of the approach, the researcher is an observer of the situation. Research output is still generated but the researcher may not develop general skills.

Another criticism of Case Study research is that it would appear to be difficult, as an observer, to gain a detailed understanding of the operation of the problem situation.

Researchers are also often teachers, skills must be learned before they can be passed on and Case Study research would not appear to support this.

As described in section 1.2 the Total Technology Scheme is an attempt to generate research output by being directly involved in solving problems faced by industry. The purpose of this is to provide a practical and general education to a researcher.

The Total Technology Scheme as applied to industry based research would, therefore, appear not just to lend itself to the Action Research cycle but to actually require application of the Action Research cycle in order to furnish the researcher with both a practical and an academic education. Drawbacks of the scheme are addressed by directly involving University and Industrial staff in a collaborative project and assisting objectivity by providing objective assistance to the researcher.

Action research was selected as the research methodology. The author was to be directly involved in systems development activity, in order to gain a detailed understanding of the problem situation before developing and testing research output.

1.9 Organisation of the Remainder of the Study

The thesis is organised into 5 main chapters. Following on from this chapter, Chapter 2 will outline some of the MPC philosophies in current use, providing greater detail on Manufacturing Resource Planning, as this is the technique used by the collaborating company. Chapter 2 will also outline the development of computer technology before describing some Systems Analysis and Design methodologies. Chapter 2 will end by discussing the findings of an investigation into documentation techniques and protocols.

Chapter 3 will then explain the development of quality systems within the collaborating company and propose a methodology for structured MPC systems development. It will continue by showing that documentation is of concern to manufacturing companies and will explain the development of a documentation system for the collaborating company. The methodology will be expanded to include the systematic control and maintenance of MPC system documentation.

Chapter 4 will then present the results of applications of the methodology and structured interviews carried out to help validate the methodologies presented in Chapter 3.

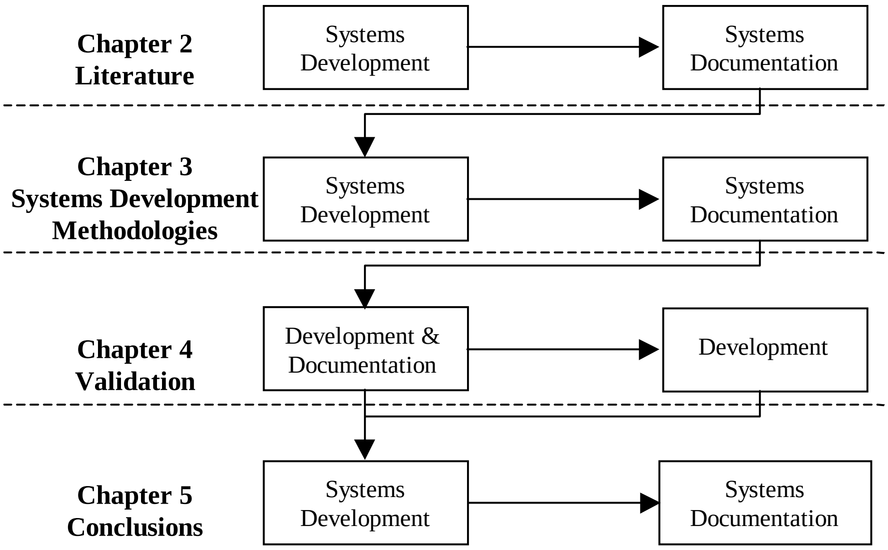
Chapter 5 will finally discuss the work presented and will draw conclusions, before suggesting further work.

There are two main themes to the work contained within this thesis:

- Structured Development of MPC Systems;
- Structured Documentation of MPC Systems.

Excluding Chapter 4 each of the above chapters is broken down into systems development and systems documentation sections. In all cases the systems development discussion precedes the systems documentation discussion. The structure and flow of the thesis is illustrated by Figure 5 below.

Figure 5: Structure and Flow of Thesis



In addition to developing and documenting the quality systems, work was done supporting many other systems within the business and implementing standard modules within the planning and control function of the business. Consulting and case study work was also done with some other manufacturing companies. The companies visited use a range of planning and control methodologies and all but one are either companies within the same group as the collaborating company or raw material suppliers to the collaborating company. Section 3.1 describes the companies visited or worked with.

The project plan for this research was developed to allow extra time to undertake this work which provided a general understanding of business operation. This is in keeping with the aims of Total Technology, (see section 1.2).

Chapter 2. Manufacturing Planning and Control Systems

2.1 Introduction

Chapter 1 detailed the origins and background of the research. The chapter showed that the research would follow a conceptual framework as the experimentation and literature search findings would be used to guide the research to a satisfactory conclusion. This chapter will start to provide this framework by providing a detailed literature review.

Chapter 2 will start by introducing some early Manufacturing Planning and Control, (MPC), systems, (section 2.2), and will then discuss the development of Manufacturing Resource Planning, one of the main MPC systems in use today, (section 2.3). Following this there will be a discussion of some of the problems found when using these systems and solutions that have been suggested, (sections 2.4 and 2.5). There are some alternatives to this approach, however, such as Just in Time and Optimised Production Technology and these approaches will also be detailed, (sections 2.6 and 2.7).

The chapter will then introduce some early developments in Information Technology, (sections 2.8 and 2.9), before detailing some Systems Analysis and Design techniques, (section 2.10) and some modelling techniques, (section 2.11). It will continue by looking at some database design, (section 2.12) and the problems faced and solutions found when integrating systems development solutions into MRPII databases, (sections 2.13 and 2.14). The chapter will then detail some current MPC developments, (section 2.15). It will continue by looking at the development of controlled documentation, (section 2.16) and the extension of paper-based documentation into electronic documentation, with associated problems, (section 2.17). The chapter will end by explaining the background of the Ferodo OE Business MPC systems, (section 2.18).

2.2 Early Manufacturing Planning and Control Systems

A Manufacturing Planning and Control, (MPC), System can be defined as a system that '*Provides information to efficiently manage the flow of materials, effectively utilise people and equipment, co-ordinate internal activities with those of suppliers, and communicate with customers about market requirements'*, (Vollmann *et al* 1997).

These systems, therefore assist businesses to control their businesses and manage the complex task of co-ordinating supply with demand with limited resources.

Work was developing on the systematic control of manufacturing organisations at the start of this century, notably F.W. Taylor developed the concept of scientific management using analytical techniques applied to the organisation of work, (George 1968). In the 1930's these techniques were being applied to inventory management and techniques such as re-order point and the Economic Order Quantity formula, (a technique for determining an optimised production batch size), were developed.

These developments can be considered to be early examples of MPC techniques. An early incarnation of re-order point theory was the two bin stock control system. The two bin system of stock control was a crude attempt to control the ordering of raw materials into stores. Two receptacles, (bins), were filled with raw materials. The bins were then used one at a time. When one bin was emptied it acted as a trigger to the purchasing staff to order more material. Bin sizes were calculated so that with average use of inventory the new supply would arrive just before the second bin was emptied.

This technique relied upon using a series of receptacles to co-ordinate material supply. An alternative approach that was developed was the re-order point, (ROP), theory, (Vollmann *et al* 1997). ROP theory suggests that using the standard raw material purchasing time and average stock usage figures it is possible to determine the minimum stock level at which new material should be ordered. This is shown in Figure 6 below.

Figure 6: Re-Order Point Theory

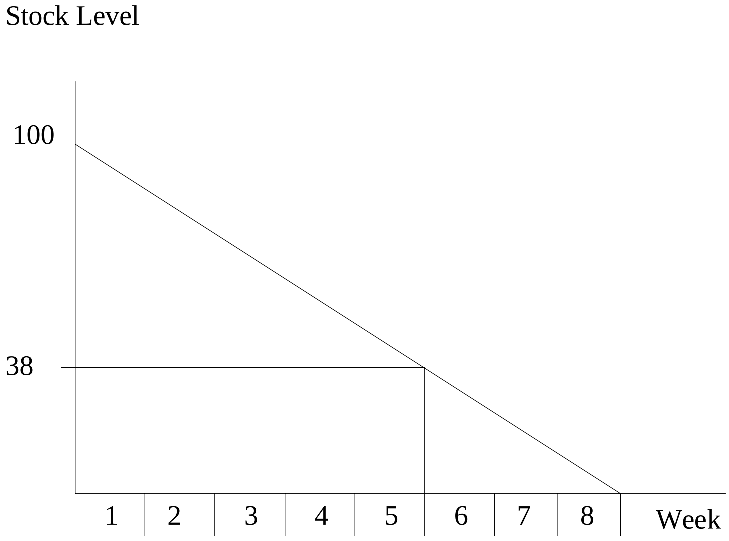
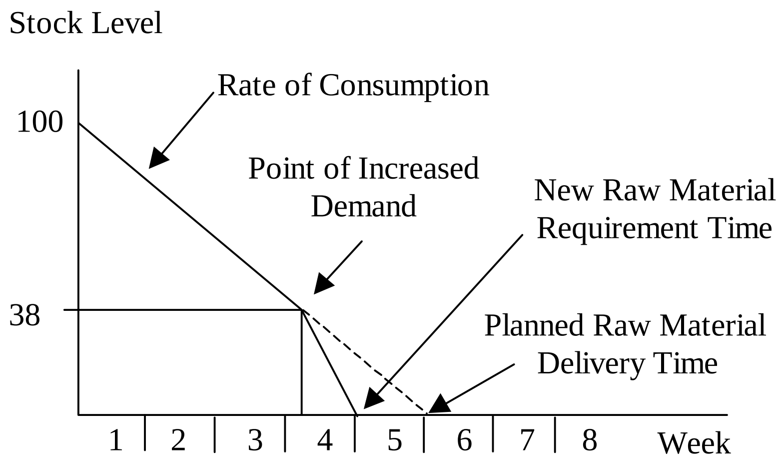


Figure 6 shows a situation where it takes three weeks to order and receive supply of the purchased material. Plotting a graph of normal usage against time it is possible to determine that 100 units are used in 8 weeks. In order not to run out of stock an order needs to be placed at the start of week 6. From the graph it can be deduced that at the start of week 6 the stock will be 38 units and thus the ROP for this particular material is 38. That is to say, when the stock reaches 38 an order needs to be placed for more material so that it arrives before stock runs out.

The strength of ROP is the simplicity of its application. There is a major weakness in that the ROP is calculated from historical consumption and takes no account of future consumption, either from forecast or actual customer demand. If the average usage was to increase then material would be ordered too late. This is illustrated in Figure 7 below where the stock is fully consumed sooner than expected and the raw material deliveries are over a week late. For this reason, an allowance, (Safety Stock), is generally applied to compensate for uncertainty in demand.

Figure 7: Effect of Increased Demand on Re-Order Point Theory



Approaches were needed that take account of future demand, either forecast or actual. During the second world war data analysis was being performed using a new technique – Operations Research, (George 1968). These techniques were applicable in manufacturing as concepts such as queuing and optimisation theory were being developed.

In addition to stock control requirements there have also been significant changes in strategy in manufacturing companies in Britain since the 1970's. During this time companies have been forced to look at mechanisms for improving product quality, costs and delivery performance. Conway, (1993), identifies a number of changes in the market that necessitate this strategy change:

- Competitive costs;
- Shorter lead-times;
- Reliability and quality;
- Increased flexibility;
- Increased product range;
- Innovation in process and product technology;
- Interesting jobs for staff.

To assist with facing these requirements companies have been forced to look at advanced MPC techniques which were developed from the Operations Research work carried out. There are any number of techniques that could be used by a company, or developed internally, to help with planning and controlling manufacturing activity. There are, however, a smaller number of well established techniques available.

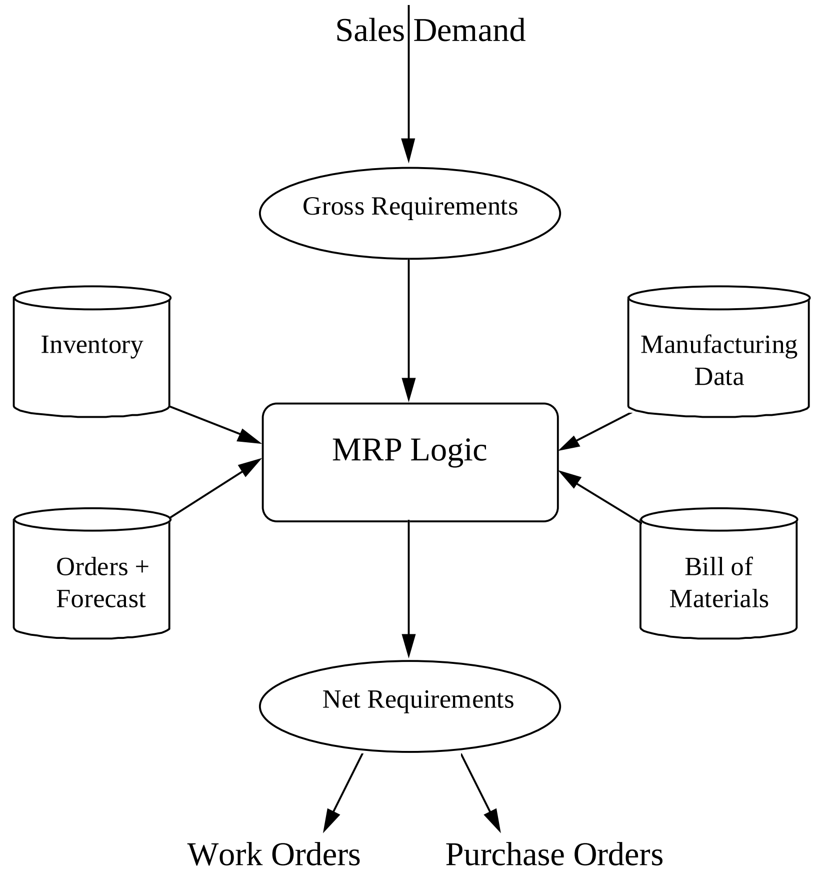
This chapter will evaluate three approaches, MRPII, Just in Time, (JIT) and Optimised Production Technology, (OPT). The techniques will be analysed, perceived flaws discussed and solutions presented. This chapter will end by showing how development of the MRPII technique led to the work presented in this thesis.

2.3 Manufacturing Resource Planning, (MRPII)

Before any exploration of MRPII can begin it is important to first to define Material Requirements Planning, (MRP) and its development into MRPII. There is a lot of confusion about the difference between these terms, for example Kamenetzky, (1985), defines MRPII as Closed-Loop MRP which Wight, (1984), defines as a level below MRPII and Fox, (1983c), defines MRP as Manufacturing Resource Planning which is commonly understood to be MRPII.

Wight was one of the first authors to write about MRP and its development into MRPII so this thesis will use his definitions of MRP, Closed-Loop MRP and MRPII. The MRP cycle is represented in Figure 8 below, adapted from Orlicky, (1975).

Figure 8: MRP Logic



MRP is a push type manufacturing system. When an order is taken from a customer, or forecasted demand is input to the system, the system, either manually or computationally, will examine the item requested and determine whether it needs to be manufactured or bought in. The system then reviews stocks and scheduled receipts of the item to see if demand can be satisfied from current and future stock. If demand cannot be met in this way, the system will make sure that there are enough items bought or made in time for demand.

This is done by examining the Bill of Materials, (essentially the recipe for a product), to determine the components that make up the item. The system then reviews each of these components in turn and determines if there are stocks or whether they have to be made or purchased. The system explodes through the Bill of Materials until the basic components are defined.

At this point, the launch date for components, (either manufactured or purchased), is determined by using the lead-time. A lead-time is the time taken for a particular process or event to happen. The system will subtract this lead-time from the customer due date and make the new date the component launch date. If the component needs to be purchased, the supplier lead-time needs to be subtracted from the component due date to determine when the order should be released to the supplier.

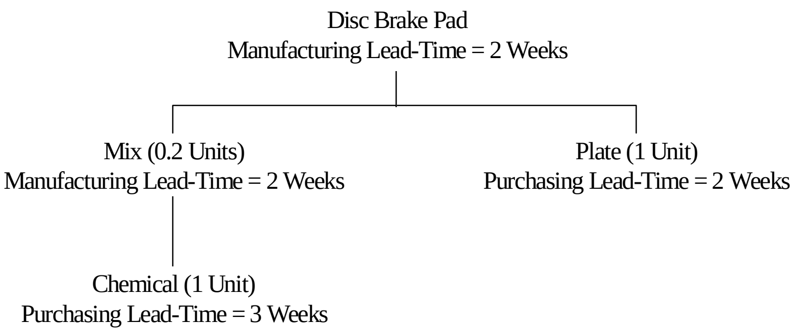
The system then releases Work Orders, (authorising manufacturing) and Purchase Orders, (authorising delivery), for components, giving enough time to process the orders through each stage in their manufacturing process, either on-site or at the supplier's premises.

Figure 9 below illustrates this procedure for a simplified Disc Brake Pad operation where a pad is made of a plate and some mix and the mix is made from a chemical. In this example 0.2 units of mix are used for every 1 plate and 1 unit of mix is made from 1 unit of chemical. It can be seen that there is a 2 week lead-time for the manufacture of both a disc brake pad and mix, a 2 week lead time for the purchase of plates and a 3 week lead-time for the purchase of chemicals. The Bill of Materials and item characteristics are shown in Figure 10 below. This example assumes no stocks, batch sizes equal to demand, no scrap and no scheduled receipts of components.

Figure 9: MRP Calculation for a Disk Brake Pad Product Structure

Week	1	2	3	4	5	6	7	8
Demand for Pad								500
Requirements for Plates						500		
Post purchase order for Plates				500				
Requirements for Mix						100		
Requirements for Chemical					100			
Post purchase order for Chemical		100						

Figure 10: Sample Disc Brake Pad Bill of Materials



In this example there is a customer order for 500 pads in week 8. Subtracting the pad manufacturing lead-time it can be seen that the Work Order for the pad must be released to the factory in week 6. In order for there to be sufficient plates and mix ready in week 6 to make the pads the Purchase Order for the plates must be sent out in week 4 and the Work Order for mixing the mix given to the factory in week 5. Working back from this it can be seen that the Purchase Order for the chemicals must be sent out in week 2.

The cumulative lead-time, allowing for this factory to procure raw materials and manufacture a pad is, therefore, 6 weeks. If a quicker delivery is required then stocks must be held or some or all of the component lead-times compressed.

This example was of an extremely simple Bill of Materials. In some manufacturing situations the Bill of Materials can have thousands of components, where some are shared between different items.

In summary, in the MRP logic the following steps are taken:

Collect gross requirements for product;

Nett off stocks and scheduled receipts;

Apply lot sizes and safety stocks;

Offset for lead-times;

Explode down to components and start again.

Before computers were used to perform this cycle it was undertaken manually. Typically it took between six and thirteen weeks to do a complete MRP run depending on the company size, (Wight 1984).

Using modern computing equipment and MRPII packages it is now possible to perform an MRP run overnight and the true benefits of the system in terms of up to date information can be exploited. For example, the MRP run in the Ferodo OE division takes less than an hour which allows it to be performed on a daily basis, (the MRP run is done during the night so as not to disrupt access to the system).

Also, with developments in computer technology it is possible to set controls on MRP, such as minimum stock levels, order quantities and safety stocks. Safety stocks of finished items can help a company deal with poor due date performance by holding stocks on site as a 'buffer' against uncertainty, (Orlicky 1975).

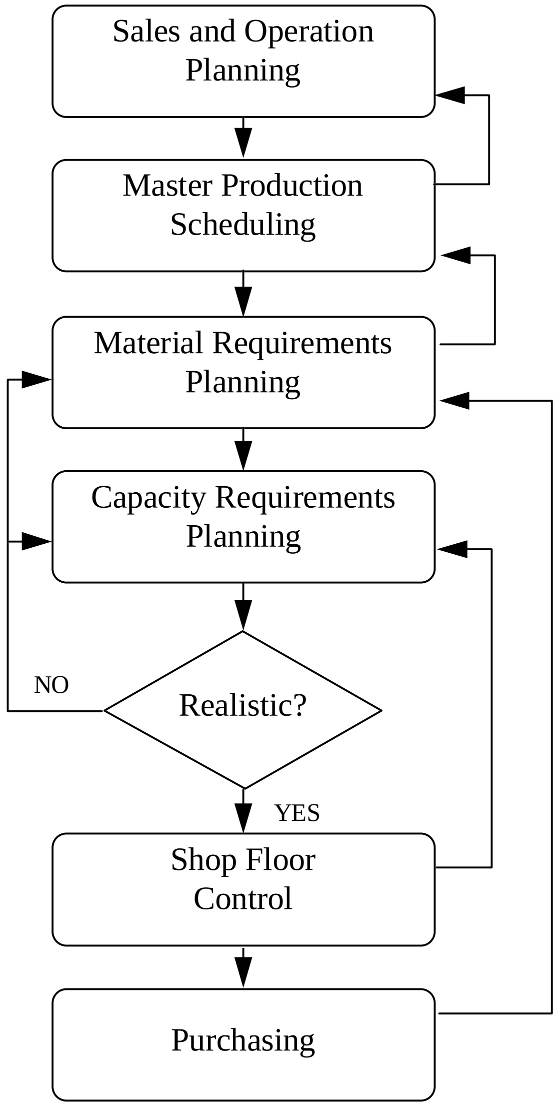
There are a number of lot sizing rules and algorithms to help calculate the 'ideal' order quantities, (Swann 1986), many of which can be used within MRP, for example:

- Fixed Order Quantity;
- Economic Order Quantity, (EOQ);
- Lot for Lot;
- Period Order Quantity;
- Set : Run Ratio;
- Least Total Cost and Part Period Balancing.

One of the drawbacks of the MRP cycle is that once the Works and Purchase Orders have been released, (given to the factory or sent to the supplier), there is no control over them and no method for monitoring their progress. For this reason Closed-Loop MRP was developed, (Vollmann *et al* 1997).

Closed-Loop MRP added a number of stages to the MRP logic so that it was possible to monitor the progress of sales orders until sales requirements are satisfied, (Wight 1984). Figure 11 below shows the Closed-Loop MRP cycle, (Wight 1984).

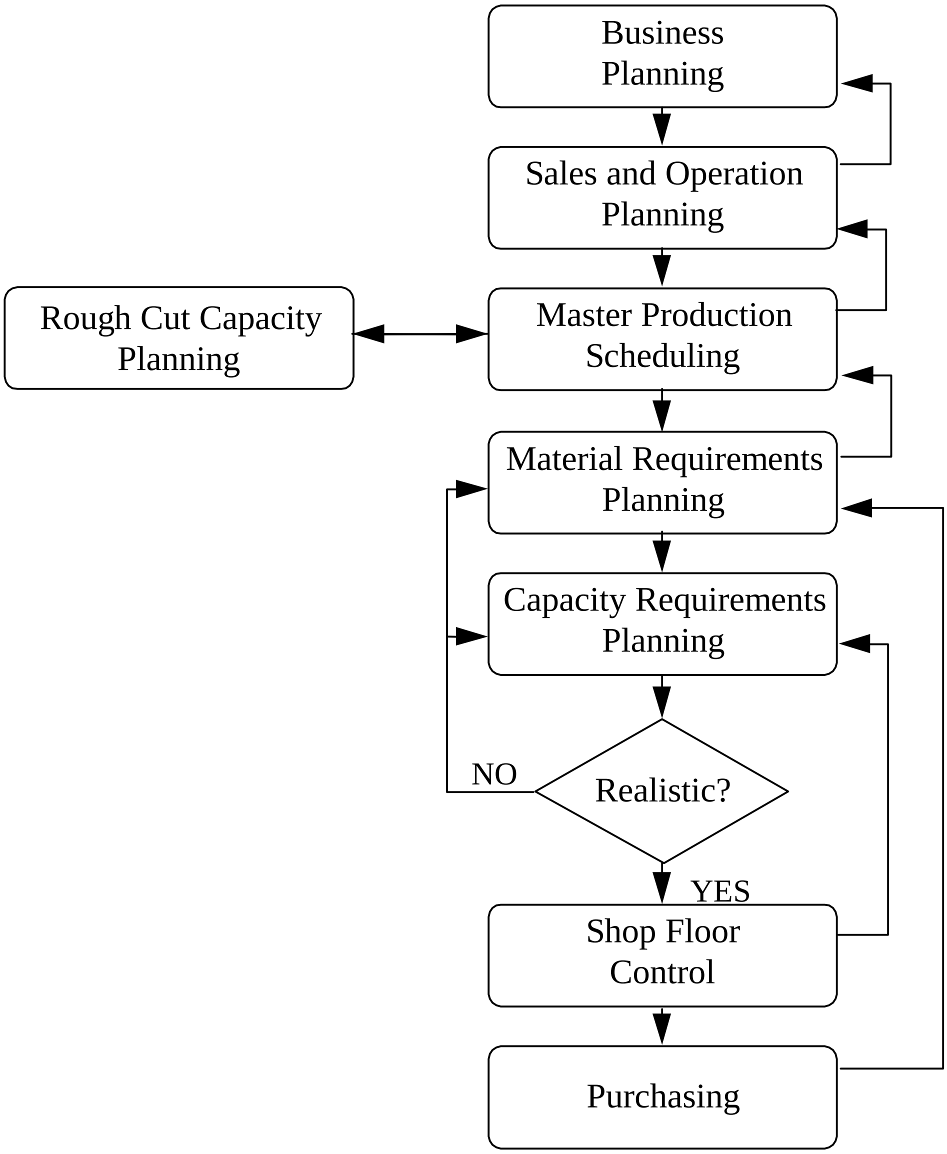
Figure 11: Closed-Loop MRP



MRPII was then developed to address more drawbacks by adding finance and accounting options, Business Planning and Rough Cut Capacity Planning to Closed-Loop MRP, (Wallace 1990).

Figure 12 below shows the MRPII cycle which is the same as Closed-Loop MRP apart from the addition of Business Planning at the top of the diagram and Rough Cut Capacity Planning, (adapted from Wight 1984).

Figure 12: MRPII



Wight, (1984), summarised three benefits of an integrated MRPII system:

1. The financial and operating systems are the same so that the accounts are a reflection of actual operation;
2. The system can perform 'what-if' analyses. Rough Cut Capacity Planning, (a tool to help with doing this), will be explained in section 2.5;
3. Everyone in the company uses the same system and can see the effect of their work.

Each stage shown in Figure 12 above represents an operation in the business with feedback between each stage. The main stages are detailed below, (Wallace 1990):

Business Planning

This is the establishment of the top level, strategic, aims and goals of the business. These aims define the direction of the business and are expressed financially. The Chief Executive and senior staff are responsible for the Business Plan.

Sales and Operations Planning

This is the weekly or monthly process of planning the rate of factory output in terms of broad product families to meet short term goals. The sales and operations plan is developed to meet the requirements of the Business Plan.

Master Production Scheduling

This is the daily process of converting Sales Orders into top level Work Orders for the items sold by the Factory. The schedule must cover the total, or 'cumulative', lead-time for the factory to allow the MRP run to calculate achievable Purchase and Work Orders. In effect this de-couples supply and demand.

Material Requirements Planning

This is the process of exploding the top level Work Orders into Work and Purchase Orders for the manufactured and purchased components as explained in detail above.

Capacity Requirements Planning

This is the stage whereby an attempt is made to link Work Orders with true manufacturing capacity. The system aggregates the total work content of each Work Order by machine or operation as defined in the routing and will predict how much capacity will be needed and when.

Shop Floor Control

This involves monitoring the actual work output of the Factory and the status of all Work Orders in the system by logging the status of each Work Order as its position in the factory changes.

Purchasing

This is the process whereby material requirements for manufacturing are met by negotiating purchase contracts using forecasts and actual demand as a guide to predict future consumption.

Due to the complexity of MRPII, systems are usually applied in the form of integrated databases of the type shown in section 1.4. Originally an MRPII database would have been written specially for an application, either on contract or in-house. More modern thinking, however, is that software should be bought in as a package rather than developed in house for the following reasons, (Cerveny and Scott 1989, Cashmore and Lyall 1991):

- There are a wide range of different packages available on the market;
- It is quicker and cheaper to buy software as a package;
- Purchased software will have been tested in the field so will be more reliable;
- The software will have been developed by skilled analysts and programmers.

Many authors have publicly acknowledged the significance of the MRPII database. Schonberger, (1984) and Kamenetzky, (1985), both state that one of the biggest contributions of the MRPII philosophy is the structured manufacturing database. Figure 13 below, adapted from Hussain and Hussain, (1995), shows a very simplified MRPII database structure.

Figure 13: The MRPII Database Structure

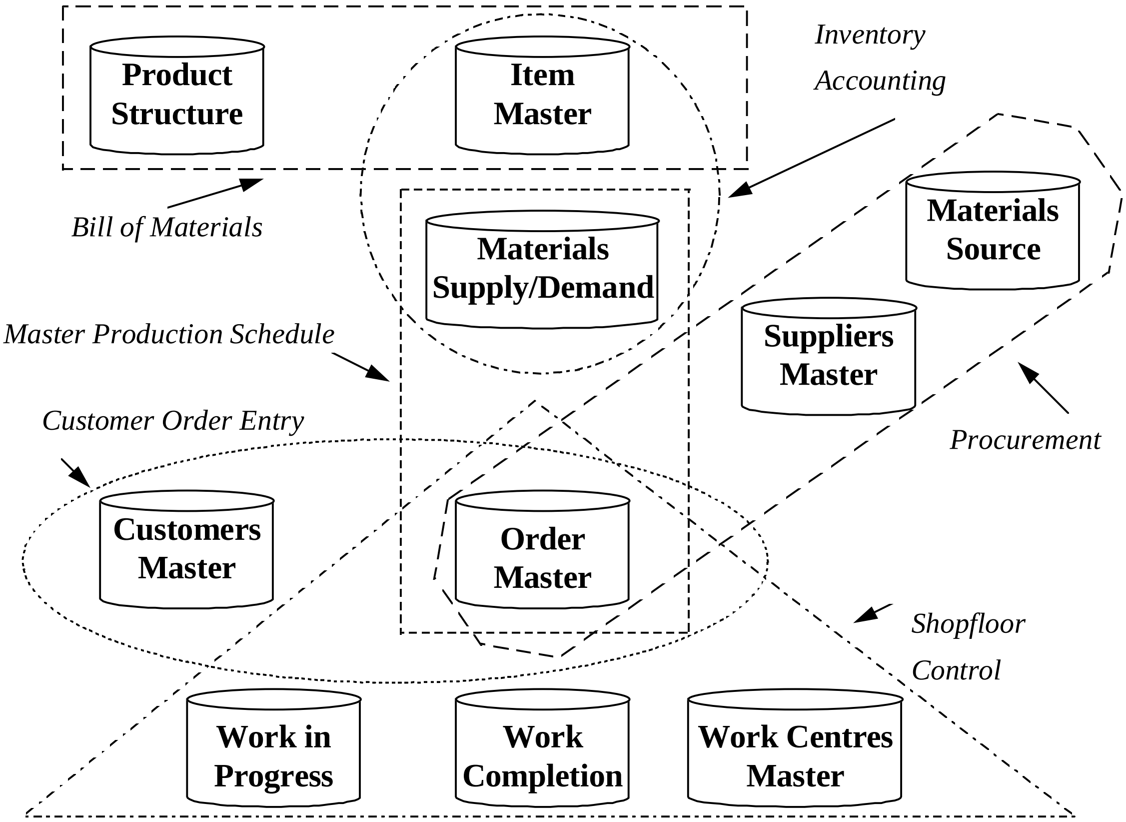


Figure 13 above shows six typical sections or 'modules' within an MRPII package:

1. Bill of Materials, (*the product structure*);
2. Master Production Schedule, (*the sequence of work for manufacture and purchase*);
3. Customer Order Entry, (*the mechanism for entering and reflecting demand*);
4. Inventory Accounting, (*stock reporting and reconciling for financial measure*);
5. Procurement, (*purchasing and contract negotiations*);
6. Shopfloor Control, (*feedback on the progress of work through the routing*).

These modules are themselves split into ten datafiles containing either static or dynamic data as below:

1. Product Structure, (*a Bill of Materials or recipe for manufacturing detailed by item*);
2. Item Master, (*a list of all items and components used, or worked on, in the factory*);
3. Materials Supply/Demand, (*a breakdown, by part, of current inventory*);
4. Customers Master, (*a list of customers*);
5. Order Master, (*a list of orders on the factory, split by customer*);
6. Work in Progress, (*a detailed list of all incomplete Work Orders in the factory*);
7. Work Completions, (*a detailed list of all complete Work Orders*);
8. Work Centre Master, (*a list of the work centres and processing times by item*);
9. Suppliers Master, (*a list of suppliers*);
10. Materials Source, (*a list of the supplier for each material and contract details*).

Because the MRPII database structure consists of a number of interlinking modules, rather than one extremely large datafile, it is possible for the same structure to be used in a variety of companies. It is not necessary for these companies to all use the software in the same way, as there are many ways each module can be used, (Orlicky 1975).

As a result of the complex nature of MRPII packages several authors have written about techniques for their successful implementation, either all at once – 'Big Bang', or in a number of stages, 'Phased'. The best known implementation technique is the 'Proven Path', originally developed by Darryl Landvater in the 1970's, (Wallace 1990), which a survey by Wilson *et al*, (1994), suggests is regarded by practitioners as the best technique. Wallace claims that anyone following the 'Proven Path' will be guaranteed a successful implementation.

To assist with defining exactly what constitutes a successful implementation, Wallace, (1990), describes a classification system with class A being the best and class D being the worst. The different classes of MRPII users can be defined generally as below:

Class D;

A class D implementation could be considered to be a failure to successfully implement and operate an MRPII package with none of the improvements shown in class C below.

Class C;

A class C company will show a reduction in inventories and will have a better ability to process engineering changes. The business would still operate as it used to but will show a very good return on investment.

Class B;

A class B company, in addition to the reduction in inventories and ability to process engineering changes from class C, will also have a dramatic improvement in the ability to deliver on time, will have minimised component shortages in the factory and will have reduced the need to work unplanned overtime.

Class A.

In addition to all measures from the class B criteria a class A company will manage the business successfully from a business plan down to the factory floor and component suppliers. These MRPII systems can be used to run what-if analyses.

The above criteria are by no means complete; there is a list of 35 questions posed by the Oliver Wight company before the classification is made and the answers are graded based on the number of questions to which a company gives the answer 'no' as below, (Wallace 1990):

0-3 questions answered negative	-	class A;
4-7 questions answered negative	-	class B;
8-10 questions answered negative	-	class C;
11-14 questions answered negative	-	class D.

The results of a survey into MRPII users showed that only half of all MRPII users were happy with newly installed systems, (Wilson *et al* 1994). In the same paper the authors state that only 9% of companies surveyed would actually classify themselves as 'class A' but that these companies showed a 200% return on investment. This figure is made more significant by the fact that the average spending on implementing MRPII packages varies little across classes A to D, (Wilson *et al* 1994).

A number of authors have claimed that a successful MRPII implementation need not use all the MRPII modules, system requirements depend on the situation being addressed, (Petty and Harrison 1995, Ang *et al* 1994). This would appear to conflict with the ABCD classification system which asks questions such as, (Wallace 1990):

- Does the company employ a forecasting system?
- Is there a capacity planning system?

An implementer without the need for forecasting or capacity planning may find that they are not considered a class A company, yet may have a good system.

Orlicky, (1975), defines three functions that an MRPII package should cover once successfully implemented and sub-divides these functions. This list does appear simplistic but would appear to provide a basis for addressing some of the operational issues that will be faced when implementing a system:

1 Inventory;

1.1 Order the right part;

1.2 Order the correct quantity;

1.3 Order at the right time.

2 Priorities;

2.1 Order with the correct due date;

2.2 Keep the due date valid.

3 Capacity.

3.1 Provide a complete load to the factory;

3.2 Provide a valid, manageable load to the factory;

3.3 Provide factory personnel with sufficient forward visibility.

In addition to operational issues, Wight defined 6 principles and philosophy changes that management should instil in a company when implementing MRPII, (Wight 1984):

1. Define the objectives of the system;
2. Assign accountability for each part of the system;
3. Develop understanding of the system through education and training;
4. Provide the correct tools for staff to perform their jobs;
5. Measure performance so that it encourages staff to work towards the objectives;
6. Provide incentives to increase performance towards the objectives.

Wight also defines the role of the Chief Executive in an MRPII implementation and the subsequent use of the package. When the package is being implemented, the role of the Chief Executive is to appoint a project team, to support them and to arbitrate but not to interfere. Once the package is in place the Chief Executive should assume responsibility for:

- The sales and operations plan;
- The production planning policy;
- Installing teamwork initiatives.

Key points when implementing packages have been forwarded by a number of authors. The most prominent piece of advice is to implement in stages either top down or in a modular fashion, (Petty and Harrison 1995, Ang *et al* 1994, Carrie and Banerjee 1984). Even the 'Proven Path' supports this method, allowing the financial modules to be implemented last, effectively creating a Closed-Loop MRP system before an MRPII system, (Wilson *et al* 1994).

This section has explained the operation of the MRPII system, which has proved to be an extremely popular system in the western world. This success has been attributed by Segerstedt, (1996), to the following attributes of an MRPII system:

- MRP considers future demand when calculating material requirements where as traditional approaches look at historical demand;
- MRP allows for components to be changed for an item and will recalculate future material requirements. This cannot be inferred from historical demand patterns;
- MRP allows for definition of a forecast based on historical demand and market intelligence rather than simply mirroring current customer demand.

There are, however, some problems with the MRP philosophy. Section 2.4 will look at some of the problems found and some solutions proposed.

2.4 Problems with MRPII Systems

At the start of section 2.3 the apparent confusion over the terminology of MRPII was illustrated. Dwyer, (1995), shows that despite the definitions of the system, many users are confused over the actual workings of MRP, Closed-Loop MRP and MRPII. There is only one solution to both these issues; education and training for all staff. Wilson *et al*, (1994), state that 80% of users need training before starting to use a system, (called 'going live'). It is apparent from experience of these systems, though, that no implementer should be satisfied with anything less than 100% of users educated before going live as skills are needed to address 'teething problems'.

In the early days of MRPII implementation, implementers were simply computerising the existing system, (Bodington 1995). This is in keeping with the data processing era proposed as the first of three computing eras in section 2.9. As a result of this the systems were likely to yield no benefits and to be considered failures, (Bodington 1995).

A further problem when using an MRPII system is that unless the data stored in an MRPII package is accurate and is kept accurate, or if it cannot be correctly extracted and displayed, it is useless, (Bodington 1995, Dwyer 1995).

An examination of methods available to ensure data accuracy must start before the implementation. Section 2.3 showed that commitment to an implementation is needed from senior management and that education is required for all staff, (a theme that continued into this section). These points illustrate that responsibility for the accuracy and therefore, the usefulness of an MRPII system rests with the users and not just with the implementation team. Dwyer, (1995), quotes an MRPII software supplier as saying that less of a system's success is the result of the software chosen than can be directly attributed to the people using and implementing the software. This illustrates that the choice of software may not necessarily be a crucial step in the implementation of an MRPII package. It is difficult to understand how this conclusion may have been reached.

Once a package is chosen and education provided it is time to implement the system. Section 2.3 showed that the most popular implementation method used was the Proven Path. Statistics have shown that many implementations fail, including Proven Path implementations, (Ang *et al* 1994, Blood 1992), in these cases failure means not fully achieving the aims and objectives of the implementation.

Published literature helps to derive a solution to this problem. At the start of Chapter 1 the policy of structured continuous improvement within Ferodo was explained. A number of authors have stated that this is vital, (Dwyer 1995, Petty and Harrison 1995) and that even a successful implementation requires continuous support to maintain valid operation, (Dwyer 1995). Statistics from Wilson *et al*, (1994), support this by showing that the classification of companies undertaking continuous improvement increases with time until an eventual class A classification is reached.

To help avoid a failed implementation it may be necessary to consider not implementing an entire MRPII package, or to implement in stages by addressing modules individually, as shown in section 2.3.

Once a package is implemented it is important to measure performance properly. In this case performance is not just of the package but also of the users. Neely, (1997), argues that in order to control the business effectively it is vital that performance measures be developed that actually reflect true performance against the strategic plans. It is important that performance measures are used to encourage staff to work towards strategic targets, such as schedule adherence. This was mentioned in section 2.3.

So far all the problems found with an MRPII package have been related to the use of the package and not with the package itself. Though significant effort can be put into addressing these problems there are some problems that some companies experience that are a function of the basic philosophy of MRPII.

Porter and Little, (1996), Kamenetzky, (1985) and Swann, (1986), discuss several problems as a result of the assumptions of the MRPII philosophy:

- MRP plans are based on infinite capacity calculations;
- MRP assumes static batch sizes;
- MRP assumes standard lead-times;
- MRP assumes standard queue sizes;
- MRP assumes a fixed routing.

It is necessary to investigate these assumptions in more detail:

Infinite Capacity.

When MRP performs its calculations it derives Work Order requirements from the Master Production Schedule. What MRP does not consider is whether or not there is sufficient capacity in the factory to process the Work Orders. There are several techniques, from the simple to the sophisticated, available to address this problem. As a result of the breadth of this area a full analysis is undertaken in Section 2.5.

Static Batch Sizes

An MRP system cannot calculate transfer batches. Transfer batches allow some of a batch to be taken to the next stage in the process before the entire batch is complete to speed up the flow of materials through the factory.

Standard Lead-Times

An MRP package cannot alter the lead-times to reflect a true prioritisation sequence through a factory. This could result in components being launched into a factory to an incorrect order start date if the process is changing or if the Work Order in question is being rushed through the factory and skipping some of the queuing time.

Standard Queue Sizes

Traditionally a queue of work sits in front of each machine waiting its turn for processing. This work is called the Work in Process, (WIP). One way to speed up order progress through a factory is to change the queue size which will reduce the lead-time. MRP cannot do this automatically, as lead-times, (and therefore, queue sizes), are fixed.

Fixed Routing

Whenever an item is configured in an MRPII package a routing is identified. This is a list of the work centres that will be used to make the item in question. MRP cannot automatically consider using another routing if there is another machine standing idle that could process the Work Order and help reduce the lead-time.

Swann, (1986), comments on these drawbacks and examines solutions. The mechanism he suggests is nothing more than human control. Swann's solutions involve monitoring and planning carefully so that overloads do not happen in the factory and manually manipulating queues and batch sizes to allow for urgent Work Orders. This solution corroborates the work of Dwyer, (1995), as presented in section 2.3, which showed that a system's success was about the people and the way they use it. The overriding message is that the users must think of an MRPII system as more than a piece of software but rather as a tool that requires human interaction. As a result, MRPII cannot act as a substitute for management.

There are, however, two technologies available that address the above issues, (these solutions are reviewed in section 2.5 and section 2.7, respectively):

- Finite Capacity Scheduling, (FCS);
- Optimised Production Technology, (which is in fact a specialised form of FCS).

It is also apparent that implementers sometimes forget that when implementing MRPII there are several other tools available for use. Dwyer, (1995) and Golhar and Stamm, (1991), argue that the concepts of Just in Time, (an alternative to MRPII), can be applied in conjunction with MRPII. These authors suggest that the implementation of MRPII systems is not simply the installation of a package but the analysis and systematic improvement of the manufacturing organisation.

The concepts of Just in Time and their use with MRPII will be covered in section 2.6 after a review of capacity planning techniques.

2.5 Capacity Planning

As explained in section 2.4 above, the capacity constraints of the factory may need to be taken into account when using an MRPII system if machine utilisation is high. This is because situations may occur where the Work Orders generated exceed available capacity.

There are five established techniques for addressing this problem considering infinite and finite capacity in the factory. These techniques are detailed as shown below:

1. Derive a Master Production Schedule;
2. As 1 but using Rough Cut Capacity Planning, (RCCP), to help the planner;
3. Use Capacity Requirements Planning;
4. Implement a Finite Capacity Scheduling package;
5. Implement an Optimised Production Technology package.

In keeping with the previous theory of encouraging user control of the system, rather than systems solutions, items 1-3 are procedures that a planner can follow. Item four is a complex packaged solution that is best applied to solve complex planning problems.

Item five warrants inclusion on this list due to its adoption of the 'Theory of Constraints' technique for addressing capacity overloads. It is, however, a commercially available package in its own right, containing its own integrated database and as such is better thought of as a replacement for MRPII rather than as an extension of an existing installation. This package therefore, qualifies as a Manufacturing Planning and Control technique and shall be covered in section 2.7.

This section will address items 1-4 of the list of capacity planning techniques in turn and will explain their use and limitations. None of the techniques, however, can predict post planning problems such as random machine breakdowns, absenteeism and sickness. These techniques can only help to identify and reschedule overloads and capacity problems that will occur due to excess demand on a factory or component shortages, (Watson 1996).

2.5.1 Master Production Scheduling

When planning production, planners can accept customer demand as indicated and attempt to launch Work Orders into the factory to exactly mirror this demand, in the same way as the MRP logic. In factories with limited capacity, if the actual capacities of machines in the factory are not taken into consideration at the production planning stage then a large queue of work will initially build up behind the slowest moving machines, (the capacity constraints) and then queues will develop behind all machines. Ultimately, customer orders will become overdue. In this case the production worker, when faced with an overloaded machine, will select the next job to do based on personal, undefined criteria which will probably not reflect the desired production schedule or any priority that the sales staff may desire to attach to jobs. This can result in expediting, where crude priority schemes mark out important jobs in the factory. Typically this will cause a complete loss in the integrity of the planning system.

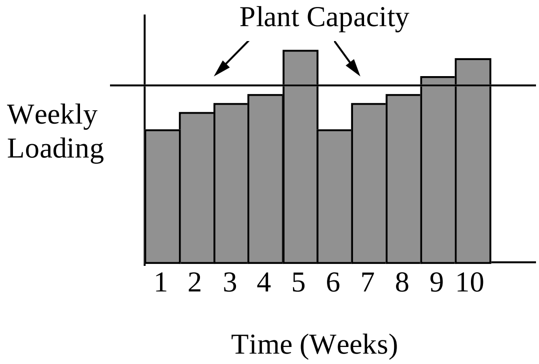
In the paragraph above the production sequence was dictated by customer orders and the production worker. It may be that this sequence is satisfactory and can be used as a Master Production Schedule, (MPS). A Master Production Schedule is simply defined as '*a statement of what is going to be produced*', (Wight 1984).

In the event of constant customer demand, then deriving a Master Production Schedule from this demand will create an acceptable schedule, (assuming that the demand is below factory capacity). If customer demand is not constant then the loading on the factory will fluctuate with customer demand. This could cause overloads on some occasions.

To cater with this type of situation the production planner must de-couple demand from the Master Production Schedule and create a ‘smoothed’ demand pattern by moving overloads into underloaded periods, (Wight 1984). Figure 14 below gives an example of Master Production Scheduling.

Figure 14: Master Production Scheduling

<u>Customer Demand</u>



<u>Ideal Master Production Schedule</u>

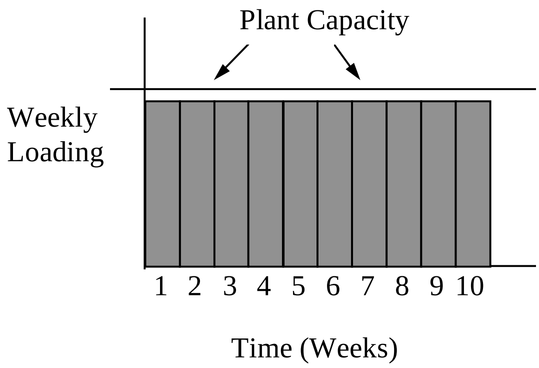


Figure 14 shows two scenarios, the first one is a schedule that reflects customer demand. It can be seen that the demand is not smooth, it actually seems to follow a 4/5 week pattern which could be linked to a monthly order cycle. In this diagram it can be seen that in weeks 5, 9 and 10 there is more demand on the factory than capacity. In these weeks there will be overdue customer orders if extra capacity is not found or overtime worked at sufficient levels to cover the overload, (if the factory is not already working 24 hours).

The second diagram shows what would happen to the loading pattern if a production planner produced an ideal Master Production Schedule. The planner in this case has taken the overload from weeks 5, 9 and 10 and spread it evenly throughout the other weeks and has allowed a bit of excess capacity each week as a safety measure. Ideally, this would be accomplished by manufacturing some of the overload early so that customer orders could be satisfied on time. It is also possible, however, that orders will have to be pushed backwards, 'planning to be late'. In this case the customer must be told that the order will be late.

The Master Production Schedule derived in this way provides a good first check on total loading on the factory. This may not be sufficient, however, as there are no guarantees that this loading will be evenly split amongst the machines in the factory. This can be illustrated by the table shown in Figure 15 below.

Figure 15: Table of Machine Capacity against Load

	Capacity	**Loading**
Machine 1	20	20
Machine 2	20	20
Machine 3	20	40
Machine 4	20	10
Machine 5	20	10
Total load on Factory	100	100
Total Possible Production for Factory	100	80

The table in Figure 15 above shows the potential drawbacks of a Master Production Schedule derived from an overview of the factory. The quoted capacity of all five machines that manufacture different items is 20 units a week giving a total capacity of 100 units a week. The total load on the factory is 100 units. In this case a Master Production Schedule derived from factory loading would indicate that the factory is within capacity. Closer inspection, however, shows that Machine 3 is overloaded by 20 units. In this scenario the factory can only make 80 units giving a maximum possible efficiency of 80%. It must be noted at this point that if production staff are measured on a factory output basis, it is impossible to achieve more than 80% productivity.

The solution in this case is to consider detailed machine loading when planning the factory and creating the Master Production Schedule. This is possible for the simple factory shown above. In an environment where there are many more machines and operations, however, the process is too complex to be done manually so there must be a compromise. This is done using Rough Cut Capacity Planning and Capacity Requirements Planning.

2.5.2 Rough Cut Capacity Planning

Rough Cut Capacity Planning is a technique available to the production planner that uses simplifying assumptions to calculate load on the factory. Once an overall factory schedule is produced, as described in section 2.5.1 above, then the same process is followed for the capacity constraints. The assumption being made is that if the capacity constraints are loaded within their capacity all the machines in the factory will be loaded within their capacity, (Fox 1983c, Wortman *et al* 1996).

If the planner discovers that there will be an overload at a Capacity Constraint the original Master Production Schedule must be manipulated until a better schedule is derived. Once the planner has derived a schedule that they accept to be good enough, they replace the original Master Production Schedule with the new one and run the MRP calculation. The next stage is Capacity Requirements Planning.

2.5.3 Capacity Requirements Planning

Using Master Production Scheduling and MRP in concert, appropriate production orders are generated. These orders can then be examined at a detailed level to derive capacity data for all workcentres in the factory, rather than just the capacity constraints. Due to the variety of routes that any factory may use, it is possible to find overloading on a particular machine. Using Capacity Requirements Planning it is possible for factory management to target the machine in question and address the problem, either in terms of overtime or extra capacity. Only in rare cases can Capacity Requirements Planning be used as part of the iterative Master Production Schedule creation process as it generally takes longer to run Capacity Requirements Planning than to run the MRP calculation, (Wight 1984).

Using Rough Cut Capacity Planning and Capacity Requirements Planning the Master Production Schedule will be as realistic and practical as possible but may not necessarily be the optimal production schedule making the best use of each individual machine. Unfortunately there are limits on the number of possible schedules that a human operator can produce and analyse. As the name, ‘Rough Cut Capacity Planning’, suggests, this type of planning is an approximating technique and as such, is inaccurate. It cannot cope, therefore, if the load on the plant is very close to capacity.

With a large number of parts, undergoing a whole range of operations, it can be seen that there can be many combinations of operations that may not be considered. For this reason, computer packages are now available to perform the above capacity scheduling process considering many more alternative schedules, optimising individual machines.

2.5.4 Finite Capacity Scheduling

Many of the several commercially available Finite Capacity Scheduling, (FCS), packages are designed to be used in conjunction with an MRPII package to help derive a workable MPS, (Roy and Meikle 1995). It is these packages that this section is referring to when using the term FCS. These computer packages are designed to be used as a tool by a planner and when fed with accurate data, will consider many different schedules in order to select the most appropriate one. The significant difference is that FCS packages will actually consider capacity at all machines when performing a schedule run and not just at capacity constraints. The requirements for these packages are, (Swann 1986):

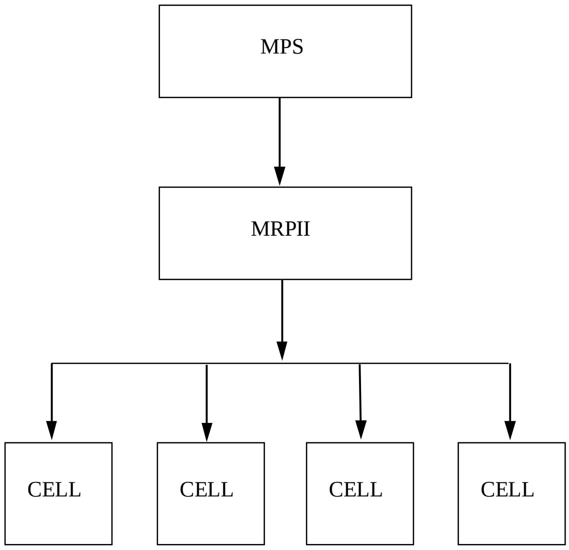
- Accurate Bills of Materials;
- Accurate Master Production Schedules;
- Accurate inventory details;
- Personnel disciplined to follow the system;
- Shop Floor data capture;
- Powerful computer systems.

The package will suggest a preferred schedule, based on user input criteria and the planner will decide whether to use this when loading the factory. If not, then another schedule will be run.

It is important not to let an FCS package manipulate the Master Production Schedule without the supervision of a planner because no system can ever be programmed with the experience of customer priorities and with the market intelligence that a good planner will have, (Watson 1996).

There are two levels at which an FCS package can be used, at the top level factory planning level, or at a bottom level operational level. Figure 16 shows a factory layout with a central Master Production Schedule driving work in four cells.

Figure 16: Hierarchical Factory Organisation Structure



Once a Master Production Schedule has been developed, the FCS program can be applied. Once an optimum schedule is reached MRP could be run to produce Work Orders for each cell. A feedback loop could be put in from the cells back to the Master Production Schedule level of the hierarchy, to give information about actual production. This could be accomplished using the Shop Floor Data Capture functions of an MRPII package.

Alternatively, the FCS package could operate at the cell level, attempting to meet the MRP schedule derived from the Rough Cut Capacity Schedule described in section 2.5.2. This does depend upon the Master Production Schedule being reasonably correct. Planning in this way can be defined as providing local planning intelligence in the cells. There is a problem in this case, when there is a shared resource between a group of cells. It is evident that, in the event of a shared resource, the capacity schedule must be run at the MRPII level of the hierarchy.

The conclusion made from this is that care must be taken when implementing an FCS package to carefully analyse the potential usage of the system so that integrity is maintained between the levels of the hierarchy.

An FCS package works using the same assumptions as a human planner. It assumes fixed process batch sizes, transfer batch sizes equal to the process batch size and average queue times, as discussed in section 2.4 above and computer systems cannot allocate overtime or bring in extra capacity, (Burgoine 1988). Kenworthy, (1994), makes the interesting point that an FCS package does not have to be better than the best human planner, only better than the average overworked planner.

Item 5 on the original list of capacity planning systems, Optimised Production Technology, does try to account for some of the drawbacks of an orthodox Finite Capacity Scheduling package. Section 2.7 will cover this after an investigation of the Just in Time MPC technique in section 2.6.

2.6 Just in Time

Any attempt made at defining the origins of the Just in Time technique must start with the work of Dr W. Edwards Deming. Deming worked for the Allied Supreme Command after the Second World War assisting with statistical control of the Japanese census. During his time with the Supreme Command, he socialised with members of the Japanese Union of Scientists and Engineers. In 1950, members of this union were reviewing a book on Statistical Quality Control published in 1931 by a statistician called Shewhart and decided to try to adopt the principles. They realised that Deming had worked with Shewhart during the war and asked him to advise them on quality control methods, (Walton 1994).

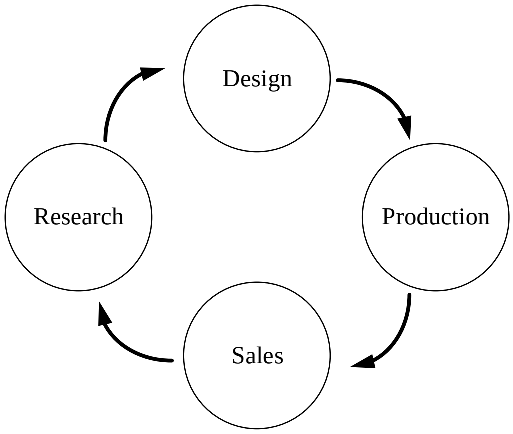
Deming set out around Japan delivering a series of seminars on quality control. Within four years of starting his lectures Japanese goods were actually in demand in the world market rather than simply being sold to the captive Japanese market. This was partly due to the success that had been achieved in radically increasing the quality of Japanese product. As a result of this success Deming built up a reputation with Japanese management. Deming used his success to start to teach the 'Deming Method' to Japanese managers, (Walton 1994, Cox 1990).

The message that Deming delivered was, (Cox 1990):

- To place an emphasis on experimentation;
- To emphasise training in statistical control methods;
- To emphasise that quality is not inspected into a product, it is engineered in;
- To have a corporate attitude of searching for constant quality improvement.

Dr Deming taught about the Deming 'Cycle' or 'Wheel', (also called the Plan-Do-Check-Action, Cycle). In this cycle, Deming conveyed the message that the entire business needs to pull together to work towards continuous improvement by constant interaction between design, production, sales and research. The Deming Cycle is shown in Figure 17 below, (Imai 1986).

Figure 17: The Deming Cycle



Building from the work of Deming, the Toyota Production System was developed, (Monden 1983). The Toyota Production System consists of several systems and methods, (Monden 1983):

- Kanban Systems to maintain JIT production;
- Production smoothing methods to adapt to demand changes;
- Standardisation of operations to attain line balancing;
- Shortening of the set-up time to reduce the production lead-time;
- Better machine layout and multi function workers for a flexible workforce concept;
- Improvement activities by small groups and a suggestion system to reduce the workforce and increase the workers morale;
- Visual control systems to achieve the automation concept;
- 'Functional Management' systems to promote company wide quality control.

At this point it becomes clear that once again there is confusion over the terminology in use. A number of authors have written in the field of JIT systems and they describe a '*JIT system*' to be a system comprising many of the elements that Monden defines as '*The Toyota Production System*', (Dear 1988, Cashmore and Lyall 1991, Golhar and Stamm 1991 and Clinton and Hsu 1997).

JIT and the Toyota Production System, (this thesis will consider both as the 'Just in Time' technique), are based on the following core philosophies, (Dear 1988):

- Develop a company wide habit of improvement;
- Develop a company wide philosophy of elimination of waste.

Just in Time is a philosophy which cannot be embodied in a software package that can simply be purchased and switched on. Everyone in the company has to work together to continuously improve the business and eliminate waste. Considering these in turn:

Continuous Improvement;

The philosophy of continuous improvement, (in Japanese, 'Kaizen'), is that all processes and systems can be improved. The philosophy also states that once all apparent improvements have been made then a new viewpoint needs to be taken, (Imai 1986).

Elimination of Waste.

Anything that wastes time, money or materials is considered to be a wasteful practice. The philosophy of elimination of waste means that all wasteful practices need to be targeted and improved, (Dear 1988).

Examples of ways in which the JIT philosophies have inspired developments in working practice will be reviewed as below, (Dear 1988):

- Kanban control of manufacturing to reduce inventory;
- Just in Time delivery;
- Reduced lead-times;
- Reduced set-up times;
- Optimised plant configuration.

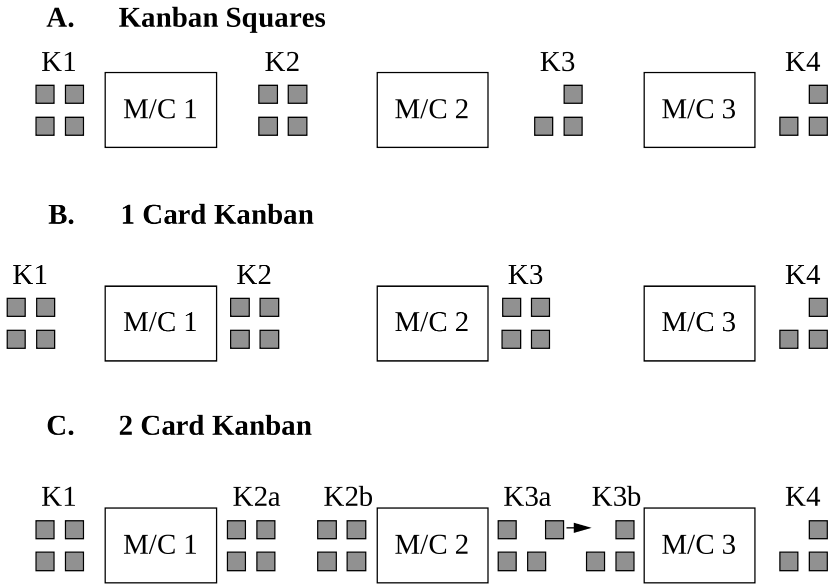
Kanban Control of Manufacturing, (Fox 1983a);

Each machine or operation in a factory converts some material or components into one or more other components. The components manufactured on one machine could be the starting components of another machine or operation. Therefore, it is possible to identify and define each type of component used at each machine. The basic concept is that no new component may be manufactured by a machine if the next machine in the routing is not ready for it. If the last machine in the routing finishes a component by processing some material from a small stock in front of it then the penultimate machine in the line gets authority to manufacture one component in order to replace the used material.

At a detailed level, there are several techniques for implementing Kanban. All of these techniques are 'pull' systems, where inventory moves as a result of the issue of material from the finished goods store. They all share the same basic control philosophy, that activity at a workcentre is triggered by the status of queues of work in front of downstream workcentres. Determination of priority is an inherent part of the manufacturing system. This is in contrast to 'push' systems. In the case of push systems priorities are determined by separate systems, such as MRP, where inventory moves as a result of components being issued into the factory, (Vollmann *et al* 1997).

Figure 18 below shows three Kanban control techniques. In the example used the Kanbans contain four units of inventory and one unit has been taken from behind Machine 3 from Kanban K4 to satisfy demand, (Browne *et al* 1996).

Figure 18: Kanban Control Techniques



A. Kanban Squares.

In this technique the Kanban takes the form of squares drawn on the ground in between machines in a line. With the gap in Kanban K4 the machinist at M/C 3 has authority to manufacture and withdraws inventory from Kanban K3 to manufacture material for Kanban K4. By doing this authority for manufacturing has been given to the machinist at M/C 2. This cycle continues until all Kanbans in the system are full.

B. 1 Card Kanban.

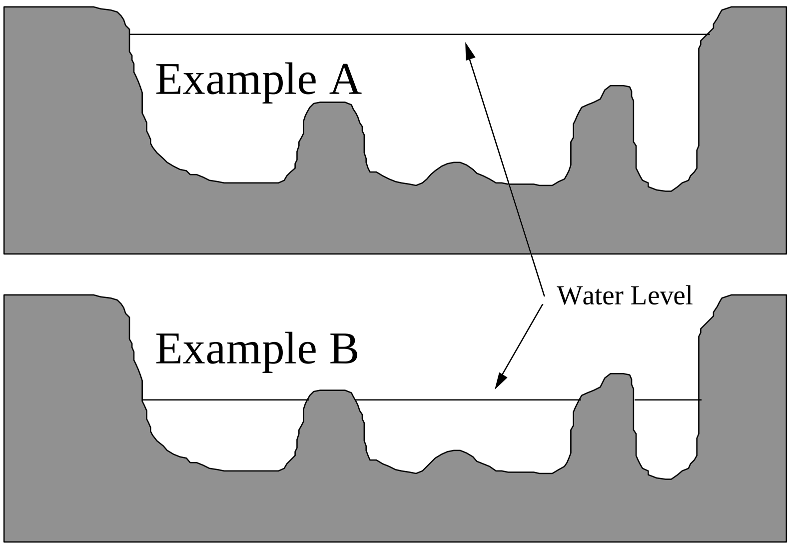
In this case the Kanban sits, as stock controlled by cards, directly behind each machine. Production is controlled by a push system, such as MRPII and material is pulled from Kanbans to facilitate production. For example M/C 3 will not fill the space in Kanban K4 until authority is given. Once it is given M/C 3 will withdraw, (pull), material from Kanban K3.

C. 2 Card Kanban.

In the final case there are two sets of Kanban controlled inventory, in front of and behind of each machine. When the machinist at M/C 3 consumes inventory there will be an ‘orphan’ card from Kanban K3b. This card authorises transfer of inventory from Kanban K3a to K3b. With this done there will be an ‘orphan’ card in Kanban K3a. This is authority for the machinist at M/C 2 to manufacture inventory by drawing material from Kanban K2b and filling Kanban K3a.

In this way work is sequenced through a factory by pulling it through the factory from the despatch area rather than detailed control at each machine with Work Orders which is the method employed by MRPII systems. As a result of this, application of Kanban is sometimes called pull manufacturing. The advantage of using a Kanban system is that once buffer stocks are controlled by Kanbans, no more inventory can build up in the factory. This approach also helps identify problems in the manufacturing system as there is less work in progress in the factory and therefore, prompts for continuous improvement, as shown by the rocks and river analogy shown in Figure 19, (Clinton and Hsu 1997).

Figure 19: Effects of Lowered Inventory



By likening a factory to a river it is possible to understand the effects and benefits of reduced stocks. Example A in Figure 19 shows a river with plenty of water, (a factory with plenty of inventory). In this case any rocks, (manufacturing problems), are covered. Example B in Figure 19 shows the effects of reducing the inventory. The rocks are exposed and by analogy, problems in the manufacturing system can occur.

There are two approaches to solving this problem, increase the ‘water level’ to an earlier level or ‘destroy the rocks’. The Japanese approach is to allocate sufficient resources to destroy the rocks, working on the principle - destroy the rock that sits highest. In this way the Japanese reduce inventory, identify problems in the factory and throw all available resources at solving the problem so that it does not halt or slow down the flow of product through the factory, (Dear 1988).

Just in Time Delivery, (Dear 1988);

Another technique that can be used to keep stocks low is to enter into a Just in Time supply agreement with suppliers. Usually a result of forming close partnerships with single source suppliers, a Just in Time agreement means that the supplier monitors the status of the inventory in the factory and supplies materials accordingly. Typically there is high delivery frequency and therefore, low delivery quantity.

Reduced Lead-Times, (Dear 1988);

By reducing work in progress and queue lengths overall lead-times can be reduced for work in the factory. Exercises at reducing the queuing necessary in a factory, (such as despatching direct from the final process), have resulted in improved overall lead-times.

<u>Reduced Set-Up Times, (Shingo 1985);</u>

Shigeo Shingo worked on reducing the time taken to configure a machine for production. He argued that set-up times could be reduced in most cases as most set-ups contain elements that could, (with planning), be done whilst the machine concerned is still running. His system, called the Single Minute Exchange of Dies, (SMED), uses two techniques to reduce set-up times and hence reduce overall lead-times:

1. Perform as many of the tasks required to change tooling before stopping machinery;
2. Improve the technology used to install the tooling to make the process quicker.

<u>Optimised Plant Configuration, (Dear 1988).</u>

There are a number of ways that a factory can be organised to optimise production. If, for example, a process calls for components to be worked on at a machine that is at the other end of a factory then logic would suggest bringing the machine closer. It may be possible to install cellular manufacturing, where all the machines needed to convert components into items are grouped together to avoid transport time, to simplify the system and to allow smaller transfer batches. Within the cell the machines themselves can be organised into a row or a U shape to create 'flow' in the cell in order to emulate the benefits of a production line, (Dale 1984, Browne *et al* 1996). It should be noted that cellular manufacturing is not universally applicable. In some cases a functional layout, (where the machines are grouped by type), may be better, for example when machine costs prohibit setting up multiple cells or when using multi-function machines.

Other elements of the JIT philosophy include, (Golhar and Stamm 1991):

- Employee participation in decision making;
- Total quality control;
- Total preventative maintenance.

With reference to the JIT philosophy, Fox, (1983b), makes the following points:

- To be able to deliver to the factory on a Just in Time basis suppliers must move their premises to be near to the factory;
- Kanban cards only work in a repetitive manufacturing environment;
- Product variety must be minimised to stop too much stock from building up in too many Kanbans, (*which results in a repetitive manufacturing environment*);
- Line stoppages must be tolerated to allow problems to be investigated;
- Resistance to the necessary culture changes must be removed, (*by education*).

It appears that many of the strengths of the JIT production system do not just apply to companies that use Kanban. Many authors have reviewed the benefits of implementing some of the JIT techniques alongside MRPII installations, (Schonberger 1984, Wallace 1990, Dear 1988, Browne *et al* 1996). Dear argues that companies using some or all of the JIT philosophy can still have high data storage requirements which require the use of an MRPII package, (1 Card Kanban also requires an MPC system such as MRPII). It is also worth noting that Kanban control of production, being derived from historical consumption, emulates the ROP technique discussed at the start of this chapter.

Browne *et al*, (1996), describe the application of some of the concepts of reduced inventory and elimination of waste when applied to MRPII installations and use the phrase, '*Lean Manufacturing*'. This does seem to be a fitting way to describe the process of continuous improvement and elimination of waste.

Section 2.7 will describe Optimised Production Technology, (OPT) and its approach to the problems of scheduling and production control.

2.7 Optimised Production Technology

In the early 1970's an Israeli Physicist, Dr Moshe Eliyahu Goldratt, applied a theory used to predict the behaviour of a heated crystalline atom to the field of production scheduling, (Meleton 1986). These ideas became known as the 'Theory of Constraints'. Goldratt formed a company, called Creative Output Limited, to write a computer package to apply these ideas. The package, sold originally by Creative Output Limited and then by Creative Output Inc. in the USA from 1979, was called OPT, (Meleton 1986, Fox 1982b). The software, in modern form, is sold today as OPT21 for large installations and as ST-Point for small installations. The UK distributors are the Scheduling Technology Group, (Softworld Report and Directory 1998).

In the same way as JIT systems, OPT works by targeting capacity constraints and reducing inventory, (Fox 1982a). Both OPT and JIT aim to help drive down costs by reducing inventory. Fox, (1983b), explains two reasons why the OPT approach is better than the JIT approach:

1. OPT is computerised rather than manual and so can perform 'what-if' analyses;
2. OPT can be applied in any manufacturing environment rather than just a repetitive manufacturing environment.

OPT is based upon the assumption that it is the goal of any company to generate cash. Normally this is represented by three financial measures, (Goldratt and Cox 1993):

1. Net Profit;
2. Return on Capital Employed, (ROCE);
3. Cash Flow.

These financial measures are difficult to relate to actual operations. Goldratt, therefore, devised the following measures as part of his OPT philosophy, (Goldratt and Fox 1985):

- Throughput;
- Inventory;
- Operating Expense.

These are defined as follows:

Throughput;

The rate of generating cash through sales. Increasing throughput, therefore, will improve net profit, ROCE and cash flow, if other factors remain constant.

Inventory;

The value of all items that will be converted into throughput. Decreasing inventory, therefore, will increase ROCE and cash flow but not net profit as inventory is viewed as an asset in orthodox accounting systems. Again, assuming other factors remain constant.

Operating Expense.

The total cost of converting inventory into throughput. It assumes all expenses are equal and does not separate direct costs and overheads. Decreasing operating expense will improve net profit, ROCE and cash flow, if other factors remain constant.

OPT focuses on the capacity constraints, which Goldratt calls ‘bottlenecks’, within a company and schedules the whole plant around all these resources, (Goldratt and Cox 1993). Once these bottlenecks are identified the Theory of Constraints can be applied.

An explanation of the Theory of Constraints compares a factory with a troop of marching soldiers. The theory argues that the slowest soldier in the group should dictate the pace of the rest of the troop. In the troop of soldiers analogy the slowest soldier would be placed at the front of the section, (Goldratt and Fox 1985).

The Theory of Constraints shows that when all machines in a factory are running at maximum productivity they will not be producing goods at the same rate. The non-bottleneck machines will be operating faster than the bottleneck machines and this will cause problems by allowing WIP to build up, increasing inventory and reducing the rate of creation of cash.

By using OPT the whole plant is scheduled to a Drum, Buffer, Rope system:

The Drum;

The drum for the factory is set at the speed of the system bottleneck, no resource may exceed this speed.

The Buffer;

The buffer is a safety stock configured to cope with problems with non-bottleneck machines. The buffer ensures that the bottleneck is not 'starved' of work.

The Rope.

The rope is the mechanism that pulls the material into the system.

By doing this, the OPT software, (using the Theory of Constraints), will ensure that the factory is not drawing in excess material. The system operates around the nine rules of OPT as defined by Goldratt, (Fox 1984), which a company using OPT must adopt as philosophies:

1. Balance flow not Capacity;
2. The level of utilisation of a non-bottleneck is determined not by its own potential but by some other constraint in the system;
3. Utilisation and activation of a resource are not necessarily the same thing;
4. An hour lost at a bottleneck is an hour lost for the total system;
5. An hour saved at a non-bottleneck is a mirage;
6. Bottlenecks govern both throughput and inventory;
7. The transfer batch may not and at times should not, be equal to the process batch;
8. The process batch size should be variable, not fixed;
9. Capacity and priority should be considered simultaneously, not sequentially.

Fox ends with the motto, '*the sum of local optimums is not equal to the global optimum*'. This is sometimes referred to as the tenth rule.

In addition to constraining the flow of work through a factory, (emulating JIT control in any manufacturing environment), OPT also contains an advanced capacity scheduling capability, (Wu 1992). OPT can schedule systems in the same way as the Finite Capacity Scheduling systems explained in section 2.5. In addition to this, however, OPT will optimise batch sizes, manipulate inventory levels resulting in reduced queue times, consider lead-times to be flexible, (by manipulating queue sizes) and allow for alternative routings, (Swann 1986). In addition to these in-built rules OPT systems will allow the user to define the criteria for the best schedule, for example due date performance or minimised production costs, (Wu 1992).

OPT performs all the tasks above through use of three elements, (Lundrigan 1986):

1. **Serve;**

   Serve is the OPT equivalent of the MRP scheduling module. In addition to the MRP logic, however, this file also manipulates batch sizes.

2. **Split;**

   Split is the file that controls time buffers and identifies bottlenecks, (machines and human resources), so the whole factory can be scheduled together to one 'Drum'.

3. **OPT.**

   The OPT file contains the rules and data required for running the factory and it also contains the OPT algorithm patented by Goldratt to apply the Theory of Constraints. The data storage is similar to the MRPII database structure, (McManus 1987).

The elements above are supported by a tool that models the manufacturing system, orders and inventory. This is called Buildnet, (Meleton 1986), which in effect combines the MRPII routing, Bill of Materials, inventory control and ordering systems.

The system is run on a day to day basis by running the Serve logic and producing a number of user defined printouts for utilisation analyses and factory scheduling. If there are errors in the schedule, (over 100% utilisation at a resource), then the data is sent to the Split module which then works out where each bottleneck is and splits work out amongst the available resources. The output from Split is sent on to the OPT file and a Finite Capacity Schedule is developed for the bottlenecks. Based on this the Serve module then creates a schedule for the non-bottlenecks. If there is still an error the cycle is performed again until the schedule is deemed 'optimised'. The average number of iterations required is five, (Meleton 1986).

McManus, (1987), reports six benefits that have been found by users of OPT software:

1. Significant increase in return on total assets;
2. Significant increase in net cash flow;
3. Significant increase in cash flow;
4. Significant increase in manufacturing efficiency;
5. Significant increase in stock turns;
6. Significant reduction in inventory.

There are, however, a number of issues with OPT, (Swann 1986):

- Data accuracy is still vital;
- Users must be taught to work towards schedule adherence rather than performance measures such as utilisation;
- Expensive computer hardware is required.

The author has reviewed an OPT implementation, where it was observed that use of OPT requires a high degree of technical expertise from the support staff and that the implementation of the software is time consuming and difficult.

Swann, (1986), identifies how an MRPII system could be used to emulate OPT. The solutions proposed hinge on interaction from a Master Production Scheduler within the planning function to manipulate batch sizes, queue lengths and due dates manually and use of a Finite Capacity Scheduling program. Roy and Meikle, (1995), show that there has been increased interest in MRPII with Finite Capacity Scheduling since OPT was developed. Meleton, (1986), suggests a reason for this being because of the mystery surrounding OPT.

It appears that because Goldratt has made the algorithms used by OPT software a commercially sensitive issue there is little understanding of its true operation amongst potential users. To quote Meleton, (1986), '*one must pay up to $500,000 for a system whose operation is a mystery and hope that it works as claimed*'. MRPII and Finite Capacity Scheduling systems are well documented and as such it is easier to justify, or not, the capital expenditure that is necessary to adopt one of these systems.

It is also worth investigating affiliations. Fox was a Vice-President of Creative Output Inc. and wrote a series of articles about OPT shortly after the company started selling it, (Fox 1982a, 1982b, 1983a, 1983b, 1983c, 1984). As a result of the association of Fox with Creative Output Inc. a criticism that can be levelled at the papers is that there is a risk that they are subjective rather than objective.

It would appear from the literature that many of the established texts on the subject of OPT are attributed to people with an interest in the product, especially the books and papers of Goldratt himself, (Goldratt and Cox 1993, Goldratt and Fox 1985).

Golhar and Stamm, (1991), attempt to identify the better system, MRPII, OPT or JIT systems. In their survey they found conflicting opinions about the use of the packages and claim that the best solution is one that best fits the situation being addressed, (which is perhaps intuitive). They go on to say that the best course of action is to apply whichever approach is most appropriate in each individual element of a situation. They argue that MRPII and JIT are particularly complimentary philosophies.

2.8 Development of Information Technology

To facilitate the predominantly computer based MPC systems Information Technology, (IT), is now widely used in manufacturing organisations in a variety of applications. It is important to define the term IT because IT and Computing are commonly confused, (Porter and Millar 1985). IT is a term used to describe systems that collect, process, store, transmit and display information, therefore a succinct definition may be '*The convergence of two technologies that had traditionally been separate: computing and communications*', (Kempner 1987). Computers are just one of the tools available to assist with IT, however, books and paper could be considered to be other tools, (Avison and Wood-Harper 1990). The field of Information Systems is concerned with the combination of Information Technology and Management, attempting to successfully solve business problems with the application of IT, (Checkland and Holwell 1998).

This section will briefly show how computers developed and became a powerful tool that assisted with the spread of IT techniques throughout businesses and will analyse in more detail the development of Systems Analysis and Design tools.

2.9 Computer Systems Development

It was in the 1950's that computers started to be used in industry in the form of large mainframe computers. As the processing speed and data storage capabilities of computers increased so did the potential applications. By the 1970's computers had developed into mini-computers capable of far greater performance than early mainframes, (Ward and Griffiths 1997). Since the 1970's the capability of computers has still been developing considerably and continuously.

Modern desktop computers, boasting high speed processors and plenty of Random Access Memory, (RAM), have significantly greater performance than the 8086 based machines with RAM measured in kilobytes available in the 1980's.

Hussain and Hussain, (1995), ascertain that the cost of computing drops by twenty percent each year. Porter and Millar, (1985), showed that the time needed for the processing of a electronic operation fell by a factor of 80 million between 1958 and 1980. There is no indication that this development will slow in the near future.

During the time that computing has taken to develop into what we know today there were three identified eras, (Ward and Griffiths 1997, Porter and Millar 1985, Reynolds *et al* 1997). These eras were:

Data Processing;

In this era the focus of computer use was to automate existing data capture, storage and analysis practices.

Management Information Systems;

In this era computer systems providers developed methods of providing and analysing information for management, to support decision making.

Strategic Information Systems.

In the final, current, era, computer systems are being used to allow implementers to re-design the business to increase competitivity in the market.

There are a number of drawbacks with current computing technology, however. Two of these limitations are, (Hussain and Hussain 1995):

- Computers lack intuition;
- Computers lack creativity.

Both of these limitations mean that humans are still required to analyse any situation and decide upon appropriate solutions to problems.

In addition to the above mentioned drawbacks it should be noted that whilst the performance of computers has been developing the software that could be run on these computers has also been improving and becoming more complex.

This complexity led to recognition of the need for Structured Systems Analysis and Design methodologies, (Ward and Griffiths 1997).

Section 2.10 describes the development of general Systems Analysis and Design methodologies and their application in the workplace.

2.10 Systems Analysis and Design Methodologies

A definition of a Systems Analysis and Design methodology has been offered by Friedman and Cornford, (1989). They explain that the word methodology is often confused with 'method' or 'technique'. They go on to attempt to clarify the situation by describing a methodology as '*a structuring of the systems development process*'. There are a number of 'structured systems development processes', this section will attempt to describe some of the better known ones.

It could be argued that the first Systems Analysis and Design methodology was the scientific method. This is a very simple methodology, adopted by students and researchers world-wide. The four steps are as below, (Wight 1984):

1. Observe;
2. Hypothesise;
3. Test the hypothesis;
4. Modify the hypothesis until it is proved correct.

Hussain and Hussain, (1995), describe five generations of Systems Analysis and Design methodologies. It is clear from their descriptions, however, that many of the early methodologies they catalogue were designed to assist with the creation of computer programs or studying problems.

An early definition of systems analysis, notably using the word 'elements' rather than computers, is '*the comparison of enlarged systems of interrelated elements*', (McKean 1958). It is clear from the above definition of systems analysis that there will be situations where the system being investigated is not based on a computer system. There is, therefore, a need for some generally applicable Systems Analysis and Design methodologies.

Four such early methodologies are the RAND methodology, Hall's methodology, Jenkins' methodology and Checkland's Soft Systems approach. The following descriptions review the main points / top level of the methodologies but do not expand the methodologies into minute detail as this is beyond the scope of this thesis.

The RAND methodology was created in 1946 by the United States Air Force, (Hall 1962). The methodology, described by Wilson, (1990), appears similar to the Scientific Method but includes analysis of the data collected experimentally. The RAND methodology is shown below:

1. Formulation, (*creation of hypothesis*);
2. Search, (*perform research*);
3. Evaluation, (*examine results*);
4. Interpretation, (*draw conclusions from results against the hypothesis*);
5. Verification, (*conduct experiments to further test the results*).

According to Wilson, (1990), a drawback of the RAND methodology was that it described the steps to take but did not attempt to explain how to take those steps.

The work of Hall and Jenkins in the 1960's, (Hall 1962, Jenkins 1969a, Jenkins 1969b), led to the creation of two important methodologies. Hall's methodology, (Hall 1962), is an attempt to convert the previously existing methodologies and apply them to the field of systems thinking. The methodology was as follows, (Wilson 1990):

1. Problem definition, (*define the system*);
2. Choosing objectives, (*decide on the new requirements*);
3. System synthesis, (*select and model the new systems available*);
4. Systems analysis, (*compare the available systems against objectives*);
5. Selecting the 'optimum' system, (*choose the best fit solution*);
6. Planning for action, (*sell the new system to future users*).

The Jenkins methodology was a major step forward in the field of systems thinking. As the first Professor of Systems Engineering at Lancaster University he had his research students apply the Action Research Cycle, as defined in section 1.8, to real life systems problems, (Wilson 1990). The resultant methodology appears in two papers published in 1969 describing the methodology and an application of it, (Jenkins 1969a, Jenkins 1969b). The methodology is as below:

1. Systems analysis, (*identify the problem*);
2. Systems design, (*develop a solution*);
3. Systems implementation, (*implement the solution*);
4. Go live, (*start using the solution*).

Analysis of these systems, with the benefit of hindsight, leads to the conclusion that these systems were targeted by their authors at real life, physical systems. The fact that working practices could also be defined as systems was not recognised until later, (Wilson 1990). In addition to this it has now been recognised that these methodologies conformed to the ‘Hard’ systems approach to problem solving, (Wilson 1990).

Checkland, (1981), introduces the difference between ‘Hard’ and ‘Soft’ systems approaches. In simple terms, a Hard systems approach is one in which a pre-determined goal is sought. The problem is defined in terms of an expected result and the systems are developed in order to achieve this result, (Checkland and Scholes 1990).

The Soft systems approach is one which has developed from work at Lancaster University with Action Research and is sometimes referred to as ‘Checkland’s Methodology’. It became obvious that real world problems could not always be defined in terms of goals and expected results. In some cases it was necessary to study the problem and in the course of that study attempt to arrive at a best fit solution. The phrase ‘attempt to arrive at a best fit solution’ is used because in practice it is difficult to know if the solution advocated is really the optimum. This methodology is well documented by Checkland and other authors, (Checkland 1981, 1991, Checkland and Scholes 1990, Checkland and Holwell 1998, Wilson 1990).

The Soft Systems methodology is shown, in simplified form, below, (Checkland and Scholes 1990):

1. Exploration of a perceived problem situation including its social and political nature;

   which leads to

2. Selection of relevant systems of purposeful activity and model building;

   which enables

3. Structured exploration of the problem situation using the models;

   which yields

4. Knowledge relevant to improving the problem situation and accommodations enabling actions to be taken;

   which leads to

5. Action to improve the problem situation.

To assist with defining a system model Checkland developed the acronym CATWOE. He argues that any definition of a system must contain, (Checkland 1981):

C Customers, (*the client or object of the main activity*);

A Actors, (*the entities that enable the system change*);

T Transformation, (*the process of the Actors changing the Customers*);

W Weltanshauung, (*the perceived view of the system*);

O Ownership, (*the ownership of the system and its management*);

E Environment, (*the environment the system is in and the restrictions imposed*).

This list of methodologies is by no means complete. For example the CATWOE definition can be seen to be a development of an earlier attempt to define the key individuals in a system development exercise called the Systems Realisation Process. Swamson, (1982), describes three individuals in the Systems Realisation Process, the designer, the decision maker and the client. The Systems Realisation Process has three stages; design, implementation and use.

A tool has also been developed to help to structure the Systems Analysis and Design process and is available as a software package. This tool is called the Structured Systems Analysis and Design Method, SSADM. The tool was developed in the 1980's in the UK and has three phases, (Downs *et al* 1988):

1. Feasibility, (*this is not a compulsory part of the SSADM technique*);
2. Analysis;
3. Design.

The three phases are themselves split down into a number of steps, (Downs *et al* 1988):

1 Feasibility;

1.1 Problem Definition;

1.2 Project Definition.

2 Analysis;

2.1 Analysis of the current system;

2.2 Specification of the required system;

2.3 Selection of service level for the new system.

3 Design.

3.1 Detailed data design;

3.2 Detailed process design;

3.3 Physical design control.

The SSADM software tool provides the detailed project management required by the Systems Analysis and Design process and includes the ability to produce graphical relationship diagrams in the form of entity relationship, (e-r), diagrams, dataflow diagrams and entity life histories, (some e-r diagram techniques are reviewed in section 2.11). It can be seen that the SSADM software computerises the Hard Systems approach to Systems Analysis and Design.

Avison and Wood-Harper, (1990), criticise SSADM as being far too prescriptive and have developed an alternative, called Multiview. The Multiview methodology has five stages and brings a new issue into the field of Systems Analysis and Design, the interaction of the systems with the eventual users.

According to Avison and Wood-Harper, (1990), the five stages of the Multiview methodology are:

1. Analysis of human activity, (*develop a conceptual model of the proposed system*);
2. Analysis of information, (*develop a functional model and an entity model*);
3. Analysis and design of socio-economic aspects, (*get feedback from eventual users*);
4. Design of the human-computer interface, (*design the user interface based on 3*);
5. Design of technical aspects, (*design the system structure to achieve 4*).

Avison and Wood-Harper claim that this methodology allows the systems analyst to develop the system through each stage, where each stage leads directly on to the next. They also claim that the methodology is effective because it allows the systems analyst to start with a general definition of the problem and to then move on to specifics. Each stage of the Multiview methodology has tools and techniques defined to allow the systems analyst to achieve the best results.

Like SSADM, the Multiview methodology allows for jobs within the design process to be split up but also cross-checks to ensure that there is no duplication and inconsistency, (Downs *et al* 1988).

All of the Systems Analysis and Design methodologies above offer useful, general tools. This thesis will develop a methodology for system development and will then expand it to include system documentation in a manufacturing environment using understanding of the composition of the above methodologies as a basis for the design.

As with all design projects there are a few cautionary notes that must be raised, however, when systems are being analysed and new systems designed, (Cashmore and Lyall 1991):

- Computer systems should improve manual systems not replicate them;
- New systems should be designed flexibly;
- New systems should simplify the original system, not complicate it;
- The information system should be applied as a servant to the business, rather than as the master.

In a similar vein Kautz and McMaster, (1994), have defined eight critical factors for a successful information systems implementation:

1. Antecedents to the introduction, (*earlier attempts to implement systems*);
2. Management support and commitment;
3. Project mission, (*clear definition of the aims and objectives of the project*);
4. Organisational culture, (*the attitudes of the company and the acceptance of change*);
5. Method usability and validity, (*is the new system going to work?*);
6. Education and training;
7. Monitoring and evaluation, (*allowing revision of the chosen strategy*);
8. Involvement in change, (*involve the end users to elicit commitment and enthusiasm*).

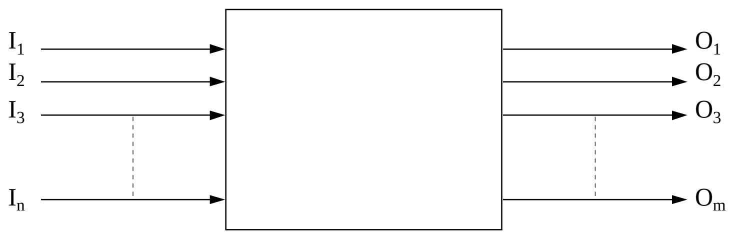
Many of the above methodologies have had a step that requires some sort of modelling or mapping of an expected system. The next section, section 2.11, will look at two techniques, Input - Output diagrams and IDEF0 and will examine their usefulness when undertaking a Systems Analysis and Design exercise.

2.11 Modelling Techniques

This section will examine two system modelling techniques, Input - Output diagrams and the IDEF0 technique. Of the two techniques the Input - Output diagram technique is simpler so it will be examined first.

Hall and Jenkins, (Hall 1962, Jenkins 1969b), both introduced the reader to the concept of the system as a box with inputs to the left and outputs to the right as shown in Figure 20 below, adapted from Hall, (1962).

Figure 20: The Input - Output Diagram



Both Hall and Jenkins explain that systems analysts need to follow the following procedure to develop a system using Input - Output diagrams:

1. For the main system all Inputs and Outputs need to be defined as in Figure 20;
2. Next the main system should be split into its component sub-systems until all inputs and outputs are included and Input - Output diagrams drawn.

It is very possible that the ratios of inputs to outputs on any particular system or subsystem will not be 1:1 and that the outputs of one sub-system could be the inputs of another.

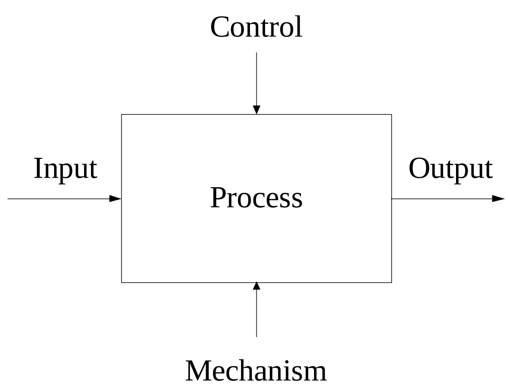
Each sub-system now should be considered to be a main system and the procedure above should be followed so that each sub-system is split into its own sub-systems.

Eventually using this procedure it will not be appropriate to form any more sub-systems and the definition procedure will be finished. The result of the exercise is that the systems analyst would have a full list of systems with inputs and outputs indicated. Using these diagrams the systems analyst can design and implement all the relevant systems.

The strength and weakness of the Input - Output diagrams technique is that it is very simple. Another technique was developed that allowed definition of controls and mechanisms for change for a system. The technique was the original "Information Definition" technique, (IDEF0).

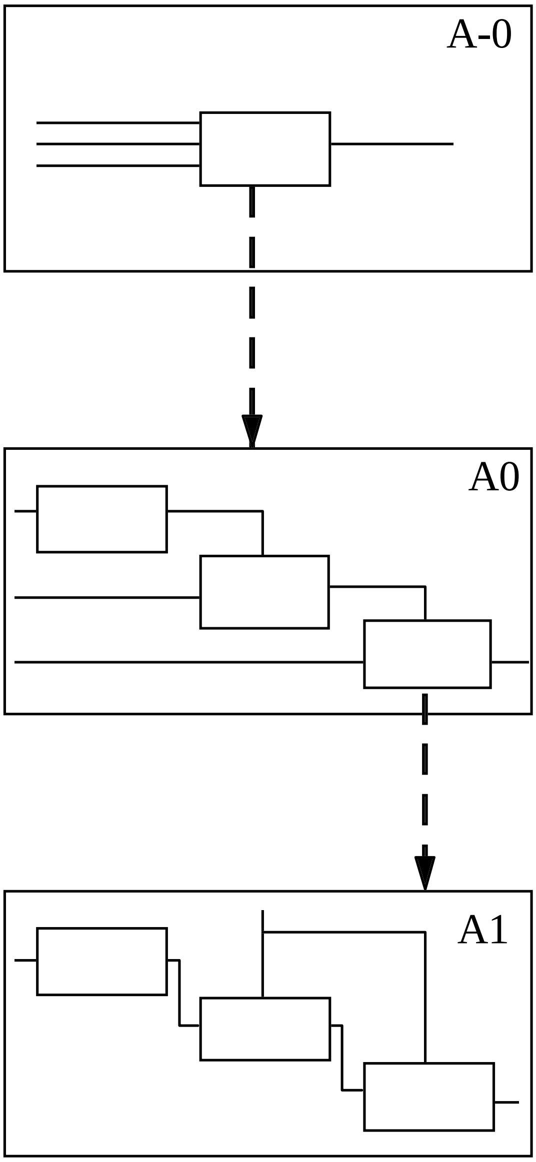
IDEF0, developed from the Systems Analysis and Design Technique, SADT, was used by the United States Air Force to model their complex integrated information systems, (Doniavi *et al* 1997). The IDEF0 model looks like the Input - Output diagram but also includes the concept of controls, (process triggers) and mechanisms, (process enablers). This is illustrated in Figure 21 below, adapted from Doniavi *et al*, (1997).

Figure 21: The IDEF0 Diagram



Just like the Input - Output technique, the IDEF0 technique considers a system to be the mechanism of converting inputs to outputs but structures the process of creation of sub-systems by a process called Functional Decomposition. This is shown in Figure 22.

Figure 22: IDEF0 Functional Decomposition



At the top level the main system is defined as system A-0. In this case there are three inputs and one output.

From the definition of level A-0 the component sub-systems are defined as level A0. In this case it is found that the output from the first stage is the control on the second and the output from the second is the control on the third, no mechanisms are needed.

From level A1 each stage is itself broken down into stages. In this case the outputs from each stage become the inputs to the next stage, the controls are now carried over from level A0. A useful check when creating an IDEF0 model is that level A0 should have exactly the same system inputs, outputs, mechanisms and controls as level A-0 and so on throughout the hierarchical structure.

There are a number of different applications of the IDEF model. Nookabadi and Middle, (1996), highlight seven current types of model available:

1. IDEF0, (*systems model*);
2. IDEF2, (*dynamic model*);
3. IDEF1X, (*data model*);
4. IDEF3, (*process description capture*);
5. IDEF4, (*object-oriented design*);
6. IDEF5, (*ontology description*);
7. IDEF6, (*design rationale capture*).

The work done in this thesis has been with information systems and as a result the IDEF0 technique has been studied. IDEF0 is the basis for Figure 21 and Figure 22.

It can be seen that both IDEF0 and Input-Output diagrams can be of significant benefit to a systems analyst. One drawback to these methods is that they are complicated to draw and need to be constantly updated by the systems analyst as systems change throughout a systems development project, (though this is made easier by use of modern software packages to draw and maintain the IDEF0 charts).

Work has been done on the effectiveness of systems modelling techniques. It has been shown from a survey conducted by Baines *et al*, (1996), that modelling is only used in 25% of all systems development projects. The survey went on to show that IDEF0, one of the more complicated approaches, is mainly used by consultancies and academia, rather than in industry.

Small *et al*, (1997b), present a nine stage process to help with a business process change project, which comprises system analysis and system design. The stages, collectively called the Manufacturing Change Framework, are as follows, (Small *et al* 1997a, Small *et al* 1997b):

1. **Diagnosis**:
    1.1. Diagnose Diagnosis, (*Formulate problem boundaries*);
    1.2. Plan Diagnosis, (*Gather data about the current system and problem areas*);
    1.3. Implement Diagnosis, (*Create and verify analysis model using data gathered*);
2. **Planning:**
    2.1. Diagnose Planning, (*Evaluate change solutions using analysis model*);
    2.2. Plan Planning, *(Design new system model with new system details*);
    2.3. Implement Planning, (*Use design model to seek company-wide approval*);
3. **Implementation:**
    3.1. Diagnose Implementation, (*Use design model to create project task list*);
    3.2. Plan Implementation, (*Produce implementation project plan*);
    3.3. Implement Implementation, (*Implement new system*).

An important step in the Manufacturing Change Framework is step 2.3, getting senior and lower level approval is vital to any change project.

The problem situations faced by the author within Ferodo were not complex enough to necessitate use of these modelling tools. It will be shown, however, that they may be applied in the systems development and documentation methodology presented in Chapter 3 of this thesis.

2.12 Database Development

Section 1.4 explained that modern MRPII software is written using relational database techniques. Cashmore and Lyall, (1991), define a database as, '*a collection of data stored in a way which allows information to be drawn from it in a range of different ways and formats to answer a range of different management questions*'.

The data in these databases is managed by a Database Management System, (DMS). These systems handle data without the need for the user to detail programs at the application level and enable data to be extracted by relatively simple means. They do this by allowing the definition of key fields to index the data and for relationships to be built up between different data stored in the database, (forming a 'relational' database). With this information defined it is possible, using the DMS, to extract and combine data in such a way that useful information is provided to decision makers within the business, (Hassab 1997).

Originally, separate databases were used by different parts of the organisation to store and analyse their own data. These databases were either specially purchased packages or solutions developed in house. According to Cashmore and Lyall, (1991), these systems had a number of drawbacks:

- They required some data to be entered many times if they were used in more than one database, which wasted time. Further to this a survey by Harrison *et al*, (1986), has shown that 68% of data errors arise from copying data between systems.
- There was no guarantee that databases were synchronised as one database may have been updated before the other. This led to lack of faith in reporting across databases.

A number of authors have described this situation as consisting of ‘Islands of Automation’, (Reynolds and Wainwright 1997, Bodington 1995, Reynolds *et al* 1997, Harrison *et al* 1995), where systems are developed independently and are used in isolation.

In the 1970’s it became evident that if the databases were somehow joined together there would be significant benefits in terms of data verification and reduced keying effort, (Harrison *et al* 1989). One problem faced was that the development of these databases had led to a number of compatibility problems due to different databases being run on different hardware, using different operating systems, (Reynolds and Wainwright 1997). Joining of databases, therefore, generally had to be done in one of two ways, linking them, (if incompatible), or integrating them, as explained below.

Linked Databases.

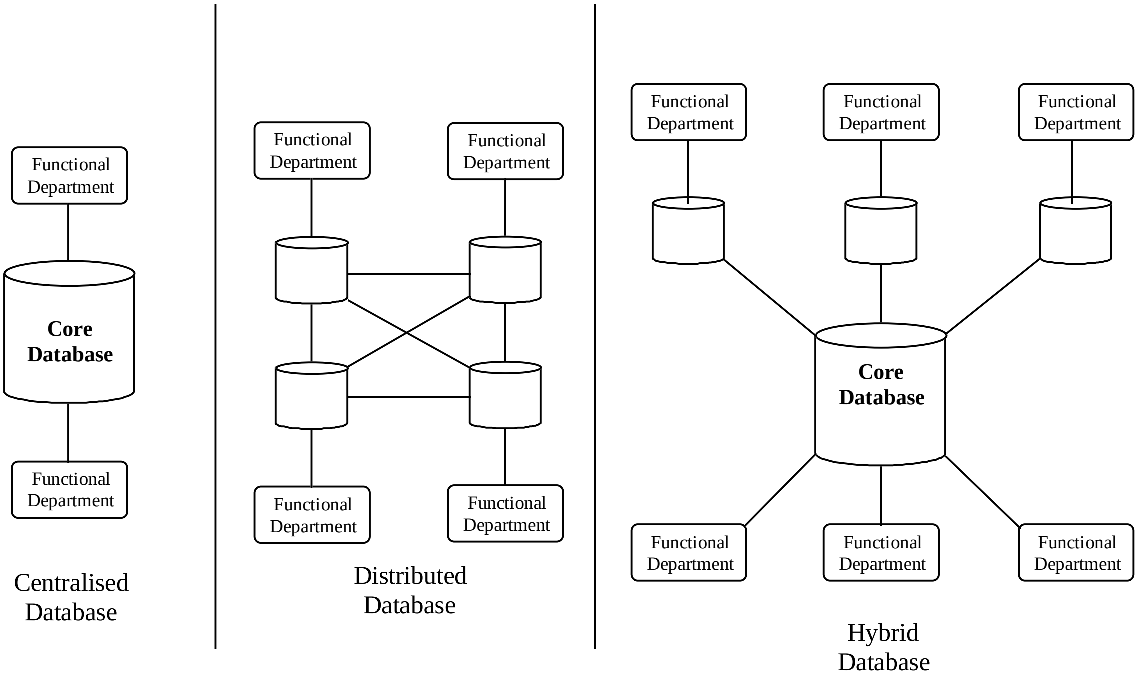
In a linked system there are two separate databases with data being sent between the two, usually in the form of American Standard Code for Information Interchange, (ASCII), files. This is usually a manually controlled process where information is sent by activating a data transfer function, (Harrison *et al* 1989). The main drawback of these systems is that the users need to know the layout of the databases to do the data transfer, (Cashmore and Lyall 1991).

Integrated Databases.

Integrated databases exist as physically separate systems that have been joined together to operate as one entity. Data is transferred freely and automatically both ways between the two systems so that the system appears to be one to the users, (McLeod *et al* 1995).

From the above descriptions it is evident that the preferred solution would be to integrate the databases. There are three approaches to integrating databases, shown in Figure 23 below, adapted from Harrison *et al*, (1995).

Figure 23: Integrated Database Approaches



Centralised Database

This type of database structure occurs when one central database is used to store and provide all the data to all the functions within the organisation.

Distributed Database

This type of database structure occurs when a number of fully integrated databases are used to store function specific data within an organisation. The DMS will ensure that the correct data is stored in the correct database. The DMS will pull data from the different databases as required and display the data as if there was one source.

Hybrid Database

This type of database structure occurs when a large database is integrated with a variety of smaller databases which only communicate with the large database and not with each other. As with the distributed database the DMS will pull data from the different databases as required and display the data as if there was one source.

One of the reasons for the success of an MRPII database is the fact that it provides an integrated approach to the data stored by a business, (Dear 1988). This integrated database structure was adapted from Hussain and Hussain, (1995) and shown as Figure 13 of this thesis. This structure is created using a number of relational datafiles integrated with one DMS between them all.

Reynolds *et al*, (1997), describe a survey that showed that respondents felt the three issues below were important to UK manufacturing:

1. Integrating systems;
2. Developing better use of IT;
3. Developing better control of the factory.

Reynolds and Wainwright, (1997), present a survey that shows that 98% of companies believe that data exchange mechanisms would greatly benefit integration in their company. They also show that 92% of respondents would benefit from a methodology to help decide which systems should be integrated for maximum benefit. The methodology they have developed has 3 stages, with each stage having individual objectives and processes before a proposal is submitted for management to consider.

An overview of the methodology is shown below, (Reynolds *et al* 1997).

Audit

In this stage, all the potential systems integration projects are investigated and mathematical models of the benefits of each system in terms of tangible figures are developed. These figures, financially represented, are reduced material cost, reduced inventory, reduced operating cost, improved sales volume and improved profit margin. The outputs from this stage to the assessment stage are total benefits, risks and costs.

Assessment

In this stage the outputs from the audit stage are analysed using Linear Goal Programming. This technique allows financial objectives to be set and combinations of projects analysed against these objectives. For example a project manager may want to get the maximum benefit with the lowest risk for a particular maximum cost. Linear Goal Programming will calculate this in terms of the outputs from the audit stage.

Suggestion

This stage allows the users to feed developed data back into the assessment stage and run what-if analyses on the data to help determine a long term strategy if, for example, a particular type of project appears to be particularly problematic or risky.

It is easy to see how the above methodology may be used to assist with justifying and prioritising integration projects with MRPII systems. This methodology does not attempt to show how to actually integrate systems once they have been approved. Little and Yusof, (1996), present the results of a study into the integration of extra modules into MRPII databases.

The study draws its findings from a postal survey conducted by the authors using data received from members of the Institute of Operations Management.

The study shows that most of the topics that it was thought would yield benefits of integration did actually show benefits. The best results were in the following areas:

- Reduced inventory;
- Reduced purchasing costs;
- Shorter manufacturing lead-times;
- Improved customer service;
- Improved responsiveness.

Surprisingly, there were some topics that were detrimentally affected by integration:

- New product introduction lead-time, (*possibly due to inefficient new systems*);
- Unexpected equipment failures, (*possibly due to extra, more complex equipment*);
- Manufacturing costs, (*possibly due to extra purchasing and maintenance costs*).

This section has established a need for integration within manufacturing databases. It has reviewed a methodology to help system designers decide upon the best combination of integration projects for maximum business benefit. It has also reviewed the results of a survey into the success of integration projects with data drawn from Institute of Operations Management members. Following on from this it is important to gain an understanding of how applications may be integrated with MRPII systems. Section 2.13 will explain some approaches to integrating applications with MRPII systems and will look at the problems faced with each approach.

2.13 The MRPII Integration Problem

Section 2.3 has already explained that there is a strong case for purchasing off-the-shelf MRPII packages rather than developing them within a company.

Wilson *et al*, (1994), present the results of a survey into MRPII systems which finds that all packaged solutions required modification of some form, most commonly by integrating other applications. Several authors have produced recommendations for the appropriate development of MRPII systems.

Wight, (1984), explains that any developed system must be designed to be simple, particularly the user interface. Wight goes on to say that the more logic that goes into a particular system the more difficult it is to understand the system and use it effectively.

Bodington, (1995), explains that a systems designer needs to create applications that can evolve as the systems develop, (ensuring a flexible design). Anon, (1995), warns that many poor systems come about as a result of computerising inefficient business practices.

Whilst the authors above show the idealised result of a systems development exercise they do not show how this can actually be achieved. Chapter 3 will review a technique devised by the author, that can be used to successfully integrate systems with an MRPII database at an operational level by using sub-systems in a hybrid database structure. This work forms the foundation of a generic methodology which will be explained in detail.

Section 2.12 has shown that there are three approaches to systems integration;

- Development of a centralised database structure;
- Development of a distributed database structure;
- Development of a hybrid database structure.

The following sections will look at these in turn and will explain how databases would be developed using each approach. It will also review the benefits and potential drawbacks of adopting each approach, (Figure 23 should be used for reference).

Before looking at integration techniques, however, it is worth looking at what constitutes a standalone system. A standalone system is a system that is neither linked nor integrated with any other system. A good definition of a situation where two systems should be left standalone would be when neither of the systems have any data in common such that data sharing would be of no significant benefit.

2.13.1 Centralised Database Strategy

A centralised database is one where all the data is stored in one large database in one physical location and it is passed as information to departments on request. Storing data in the same place yields several advantages. A centralised database removes the possibility of data duplication and reduces the chances of incorrect data input as the system can cross-reference entries. In addition to this a centralised database has the ability to create and store an audit trail as all transactions happen within the database itself, (Cashmore and Lyall 1991). The ability to create an audit trail is not unique to a centralised database but is significantly easier to achieve due to a common data store.

The problems arise when the database is large and when modifications to the structure are required. A large database will require detailed, expert, supervision and support, far more so than a small database. Wilson *et al*, (1994), show from company case studies that modifying the structure of a large database leads to problems as the database complexity causes confusion over the likely effect of changes. As a result of these problems, it is evident that changes to a large centralised database should only be made when the basic system has a flaw that fundamentally inhibits day to day business practice.

When attempting to integrate a new system into a centralised database it can be seen that the centralised database will have to be extended to include the new system logic and data storage capability. If this is done successfully the effect will be to increase the size of the centralised database and make further integration projects more difficult.

2.13.2 Distributed Database Strategy

A distributed database is a database where a number of small databases are integrated together to give the appearance, as far as users are concerned, of one system. In the definition of distributed databases given by Cashmore and Lyall, (1991), they explain that a distributed database exists over multiple locations. With modern telecommunications, however, the physical location of a database is relatively unimportant. Therefore the databases within a distributed database could exist in one physical location with each part of the organisation having control over a particular database within the distributed database, logically rather than physically splitting the databases. A better definition of a distributed database is, therefore: 'A collection of independent databases, integrated to give the appearance of a single, united whole'.

The advantages of distributed databases are, (Cashmore and Lyall 1991):

1. They transfer control of each database within the whole to departments;
2. They allow tight security as individuals can simply be barred from certain databases;
3. Individual databases can be upgraded or developed independently without the need to affect the performance of the other databases whilst doing so.

The problems with a distributed database occur when attempting to integrate a new database. To integrate four databases as shown in Figure 23 requires six connections between databases to link each one together. Each time an application gets added to a distributed database each existing application needs to be linked to the new one. The number of links n needed to connect m databases is given by the arithmetic progression:

$$n = (1 + 2 + 3 + ... + (m - 1))$$

This can be solved using the standard formula, (Stroud 1992), which can help derive n and m by substituting for y = n and x = (m-1) into the standard formula as below:

$$y = \frac{x(x+1)}{2} \Rightarrow n = \frac{(m-1)((m-1)+1)}{2} \Rightarrow n = \frac{(m^2 - 2m + 1) + (m-1)}{2} \therefore n = \frac{(m^2 - m)}{2}$$

This formula shows that the number of links and therefore, the system complexity, rises in proportion to the square of the number of databases. This will also quickly lead to complexity problems if the databases are to be held in physically separate locations. There are two ways that distributed databases can be developed to reduce this problem:

1. If the database is logically rather than physically distributed, then all the links can be implemented purely by software;
2. If the new systems are developed by integrating a new application within an existing database within the distributed database there will be no need for external links.

2.13.3 Hybrid Database Strategy

A hybrid database consists of a central large database which is integrated with smaller databases to give the appearance of one whole. The difference between this and a distributed database is that the smaller databases only communicate with the large database and not with each other. This concept is mentioned, though poorly defined, by Bodington, (1995). Bodington suggests that sub-systems should not be joined to each other but each should be joined with the main database. This suggestion does not go on to explain what a sub-system is, or how to do this joining. Doniavi *et al*, (1997), also mention the use of sub-systems to reduce complexity in systems design but again no detailed explanation is offered as to how this might be done.

It can be seen that because the integration of a hybrid database is done through the main database the number of physical links that are required is minimised, (one per sub-system). This reduction in integration complexity can be achieved whilst still gaining all the benefits of transferring control of the smaller systems to those responsible for their use and increasing the security of the systems as a whole.

It was shown at the start of Section 2.13 that packaged systems always require some sort of modification once implemented and this includes MRPII systems. The problem facing systems developers is which of the approaches to adopt when integrating new applications with MRPII databases.

The work done with Ferodo has been to apply the hybrid database approach to MRPII development to develop 'satellite systems' from the core and sub-systems approach outlined in section 2.14.

2.14 The Core and Sub-Systems Approach to Systems Development

When faced with the problems of integrating systems with MRPII systems there are a number of approaches that can be adopted. These approaches were explained in section 2.13 above and involve developing one of the following:

- Centralised databases;
- Distributed databases;
- Hybrid databases.

A modern modular MRPII package can be implemented in two main ways:

1. Implementing the modules in one place, (*physically centralised*);
2. Implementing the modules in physically separate locations, (*physically distributed*).

It has been shown that for the purposes of systems development the physical make-up of a database does not necessarily dictate the database view presented to the users. This can be illustrated by Figure 13 which showed a summarised modular MRPII database structure, as a series of integrated datafiles. The datafiles shown within this diagram were grossly simplified by Hussain and Hussain, (1995) but help illustrate the complexity of MRPII systems. Despite this internal complexity the outward appearance of these systems, however, is of a centralised database.

McLeod *et al*, (1995) and Harrison *et al*, (1995), showed that successfully implementing a 'core' system, before integrating other systems incrementally, following the hybrid systems approach to integration, reduced the complexity of subsequent system development. This can be considered to be a core and sub-systems approach to systems development.

2.15 MPC System Current Developments

There are a number of areas of development of MPC systems including:

1. Extended Enterprise / Outsourcing;
2. Enterprise Resource Planning;
3. Internet Integration;
4. Object-Oriented Programming.

2.15.1 The Extended Enterprise

The concept of the extended enterprise developed to help businesses reduce costs by focussing on certain aspects of a particular product's life cycle. This is achieved by a number of organisations working together to produce a particular product, (Browne *et al* 1996). The origins of the extended enterprise are in the concept of outsourcing.

Outsourcing is the practice of contracting external companies to undertake tasks within the business so that the business does not need to create the infrastructure to do so themselves. An example of this could be computing, where a small firm asks a specialised computer solution provider to maintain the computing facility for a negotiated fee. Another common example is premises management. The extended enterprise takes this a stage further and creates a series of companies, each with their own 'expertise envelope', working together to produce product.

A successful extended enterprise relies on good communication between companies and good information systems facilitate this. Internet technology can be used to provide this communication, this is described in section 2.15.3 below.

2.15.2 Enterprise Resource Planning

It is difficult to find a definitive text on, or definition of, the concept of Enterprise Resource Planning, (ERP). The author's experience of traditional MPC techniques has shown that these techniques assist companies to plan production of finished product within a business, it has been more difficult to plan production across a multi business organisation. Data reconciliation programs have been employed to consolidate reporting across these businesses depending upon monthly data entry.

With the advent of powerful communications technology and easily integrated systems it is possible for a large enterprise to link their manufacturing facilities and extract up-to-date information from their respective manufacturing databases. Further to this it is then possible to produce medium and long term plans from this information. Central to ERP is the concept of product families. Product families are groupings of similar products dispersed across a number of businesses.

ERP tools, (usually modules bolted onto what were previously called 'MRPII' packages), allow for definition of these product families and production of consolidated medium term production plans using historical data extracted from databases within each manufacturing plant. These plans are then broken back down into plant specific plans and distributed back to the individual plants.

The plans can then be tracked so that actual performance can be checked against planned performance to help the derivation of subsequent plans.

2.15.3 Internet in Manufacturing

The technological foundation of the Internet was developed by Tim Berners-Lee, in 1989, at the European Particle Physics Laboratory, CERN, as a mechanism for communicating research ideas between researchers in different countries, (Borälv *et al* 1994). The technology that was developed was a 'hypertext' system that was accessed using an interface called 'Mosaic', (Borälv *et al* 1994). The hypertext system was itself developed into a programming language called the Hypertext Markup Language, HTML, in an attempt to create a standard format for document storage and distribution and it efficiently uses the capacity of the phone lines or network used to do this.

Users of HTML can create documents directly in an HTML editor or they can create documents using another package, (such as Microsoft Word). These documents can contain graphics as well as textural information. Once the documents are created they can be linked together electronically using 'hypertext links'. These links, which appear on screen as highlighted text, are quick links into another stored document. Activation of a link will retrieve the document in question and display it on screen. The most significant aspect of these links are that, because of the standardised HTML format, the document in question can be on any computer that is linked to a phone line or network.

Once this technology was developed it was released for public use. There are a number of software packages now available to search for documents, 'browsing' and to create documents, 'authoring'. This technology has led to a world wide network of computers where any stored document can be linked to any other document on any other computer.

The change from paper publishing to use of this so called 'World-Wide Web' has been described as a 'Paradigm Shift', (Fryer 1996), a change so significant that it revolutionises current working and thinking. The technology of the Internet is continuously developing. Developments include Applets and the JAVA language platform, (Meckler 1997). Applets could be defined as '*small programs that display multimedia on a web-page or present simple data-entry forms*', (Meckler 1997). Simply put, Applets are a mechanism for automating tasks whenever web pages are accessed. Developing from this it has been recognised that larger web based applications can be built by combining a number of small Applets, (Meckler 1997). This emulates Object-Oriented programming, as defined in section 2.15.4.

For the World-Wide Web to be accessible to all there is a need for a Web programming language that allows users of the different PC operating system formats, (such as Macintosh or Microsoft Windows), to all access and view the same web pages. Typically these two formats are incompatible without some translation program. JAVA meets this requirement by being a programming language that can run under any operating system and allows a web page to be viewed by any operating system, (Goodwins 1999).

There are traditionally two business applications in particular that the Internet can be used to help address, communications and document distribution, (document distribution will be discussed in detail in sections 2.16 and 2.17). Internet communications are usually interpreted as 'Electronic Mail', (EMail). The internet can, however, be used by businesses to link their manufacturing databases and using JAVA they can even allow customers access to the databases to track order progress, (David 1998) or to place an order directly on the ERP system themselves, (Greek 2000c).

Recently there has been significant work done on expanding the use of the internet in manufacturing. In particular work has been done to help collaboration between different levels of the supply chain, (Pullin 2000a). Communication within a supply chain is not a new concern, as early as 1956 Jay Forrester was working on understanding and simulating the effect of changing demand upon the supply chain, (Greek 2000a). The 'Forrester' or 'Bullwhip' effect is best illustrated by example. The example given by Greek, (2000a), is as follows:

If a shop-keeper notices above average demand for a particular product type they may place an order for an extra box of the product. The wholesaler may interpret this as a likely universal effect and may place an order on the manufacturers for a larger quantity to ensure that they have enough stock to supply extra demand from all their customers. If this is replicated when the manufacturer places demand on their suppliers and further down the supply chain it can be seen that the whole supply chain could be affected by the abnormal demand. If the above average demand is a local effect then the entire supply chain may find that they have transferred production / financial resources to producing extra product that is not required, possibly at the expense of product they may have been able to sell.

It can be seen from this example that poor communication in a supply chain can potentially lead to increased stocks being held. This would tie up capital, increase the risk of obsolescence and risk production capacity being stretched / purchased at the expense of other saleable product, which would also consume capital. An example of this is silicone chip production, where an expected surge in demand did not materialise. As a result of this the computer industry found that they had a surplus of RAM memory chips which drove prices down, (to the benefit of the consumer).

This did not benefit manufacturers, some of whom had built new factories to cope with the forecast demand. Ultimately this scenario does not benefit consumers also, as in the long run prices go up due to cost recovery or businesses stopping trading, which causes short supply, (invoking the law of supply and demand).

Technology is starting to provide tools that may be used to automate the communication of this data both up and down the supply chain. If, in the example above, the manufacturer and the suppliers had access to the data that was generated by the shop-keeper when placing the order the over-supply situation may not have occurred. The tools being provided by software companies include Customer Relationship Management and Supply Chain Management programs. These programs are being integrated into the standard ERP package offerings by the ERP software authors and control the transmission and collection of data up and down the supply chain, (Pullin 2000a). Product Data Management systems are also being used to capture engineering design and production data for transmission.

These software tools are internet enabled, allowing the automated transmission and collection of data in the supply chain. Product Data Management systems, for example, are being linked to allow ‘simultaneous engineering’ of product, where different levels in the supply chain work together on the design of a final product, each company responsible for engineering the parts that fall within their expertise, (Pullin 2000b). This technology facilitates the Extended Enterprise presented in section 2.15.1.

There is a new type of ERP system provider. Called an Application Service Provider, they provide a centrally located ERP database that they lease out to multiple companies.

The benefit of this arrangement is that there is only one pool of IT staff supporting the ERP databases yet multiple companies using them. The ERP databases are accessed across the internet, (Pullin 2000b).

Despite these developments in technology the take-up of internet technology has been slow. There is still a degree of wariness over the viability of using the internet for business transactions, called E-Business, (Greek 2000b). Though 83% of firms have a website, this is generally only used as a 'shop window'. Only 24% of companies take orders on the web with only 6% generating more than 40% of sales. Only 15% of companies will accept payment on the internet, (Greek 2000c). From the customer's point of view there is also a resistance to paying for goods electronically.

In addition to security concerns another problem with the take-up of the internet is its speed. Electronic transactions can be lengthy, particularly at peak times. Communications companies are looking at ways of increasing the amount of data that can be transmitted electronically, down telephone lines or down dedicated internet lines. From a software point of view a new internet language called the Extensible Mark-up Language, (XML) has also been developed. Unlike HTML this language uses a pre-defined dictionary letting computers only transmit reference characters and the internet browser will translate these characters into a presentable format. This new language will speed the internet up, (Greek 2000b).

It is clear from the rapid development being witnessed in the area of internet technology that this technology is still in its infancy. It is also clear that the internet is 'here to stay' and that as a result manufacturing companies are going to have to develop an internet presence in order to continue to compete.

2.15.4 Object-Oriented Programming

The development of traditional manufacturing databases was described in section 2.12. A recent development in the area of software engineering is Object-Oriented programming. According to Young, (1996), '*In this style of programming, large programs are broken down into subtasks that can be combined and recombined to produce new programs*'. For example a programmer could break a particular database report down into three stages, criteria entry, data processing and data output. If these stages were programmed independently then the report would now be a combination of three 'objects'.

Accompanying Object-Oriented programming is the concept of re-use. The report breakdown described above yields three smaller programs, all of which may be common to a number of programs. If the data output program could be used to output data from a number of similar reports then the program is 're-used'. It has been shown that under unstructured, 'passive' circumstances 15% of software is re-used. If a structured attempt is made at software re-use it is possible to achieve up to 90% re-use. This can reduce the time taken to get a program onto the market to months rather than years and can increase software quality as mistakes only need to be corrected once, (Jacobson *et al* 1997).

This approach, which saves companies from having to 're-invent the wheel' is being used by a number of MRPII software suppliers to assist with database development, including the authors of MFG/PRO, (Young 1996).

Work has been done to assist with the creation of objects during Object-Oriented programming, (Jacobson 1995). Jacobson has worked on a technique which he calls use-case modelling.

Use-cases are constructed when defining the requirements for the new software. A use-case is used to assist the program developer program the object, (Jacobson and Christerson 1995). Jacobson and Christerson define use-cases and objects as follows:

Use-Case:

A use-case is the end user's view of the proposed application.

Object:

The object is the developers view of the proposed application.

Use-cases help to construct four views of the proposed application, (Jacobson 1995):

1 Definition of the system user's needs;

2 Definition of the interface to other applications;

3 Definition of project management requirements;

4 Definition of the technical authoring requirements.

A long term vision of MRPII databases programmed using Object-Oriented techniques has been expressed by David, (1998). David suggests that eventually it may be possible to purchase MRPII modules from different MRPII software suppliers and combine them so that a company may construct a customised MRPII database.

Object-Oriented programming and Use-Cases are a recent development in a series of Software Development, (called Software Engineering), approaches. The problem has always been control over the large amounts of programmed code required to produce computer applications.

For example, in the 1980's Mills *et al*, (1987), wrote about the Cleanroom approach to Software Engineering in which the goal was a reduction in the number of defects found in a program. This was achieved through defect prevention rather than defect removal and developing software in a series of smaller, incremental stages which were then independently tested and certified.

Mills *et al* describe Cleanroom as follows:

The Cleanroom process '*calls for the development of software in increments that permit realistic measurements of statistical quality during development, with provision for improving the measured quality by additional testing, by process changes (such as increased inspections and configuration control), or by both methods'*

Statistics show that the Cleanroom approach reduces the number of defects passed through the launch of a new software product by finding approximately 90% of all defects as opposed to 60%. Despite the pace of development in the computing industry Cleanroom is still applicable today, (Sherer *et al* 1996).

There is a striking similarity between the Cleanroom approach, (incremental programming) and Use-Case, (small, interacting programs). Both approaches break a large program into smaller elements to improve quality and to ease programming.

2.16 The Documentation Problem

With the development of ever more complicated computerised manufacturing planning and control systems there is an increasing need for adequate accompanying documentation. Lehner, (1993), shows that due to the complexity and quantity of software there is an increasing need for documentation to assist with post-implementation software maintenance. Lehner goes on to say that as software gets more complex the need for good quality documentation will rise.

Software maintenance is not the only driving force behind documentation. Knowles, (1995), describes a document as a medium to enable information to be passed on and understood. In industry the information can be from a variety of sources, for example:

- Customer address lists;
- System operating procedures;
- Health and safety procedures;
- Corporate policy and aims;
- Approved suppliers lists.

Knowles also shows that the management of documentation and the information contained therein forms the core of a company's ability to operate effectively. To help understand what document management entails it is worth reviewing the types of documents that will be created. Kliem, (1984), shows that there are two types of documents that are needed in a company, operational and administrative. Operational documents, (or procedures), explain how to perform certain tasks within the organisation. Administrative documents provide supplementary information necessary to carry out the operational tasks.

There are three levels of user, (Bradford 1988):

1. Novice, (*a user with a limited understanding of systems*);
2. Sophisticated, (*a user with a good working knowledge of use of the system*);
3. Expert, (*a system administrator or extremely experience user*).

All three user types require different documentation, as described by a number of authors. McLeod *et al,* (1995), suggest that there are four levels in the business at which documentation is required, each one aimed at all three user types:

1. Organisation, (*detail about how the business operates*);
2. Function, (*detail about how each department allocates work*);
3. Application, (*detail about how each individual processes their work*);
4. Module, (*detailed system operating procedures*).

Hussain and Hussain, (1995), show that for a complex system there should also be four documents, these documents are:

1. A systems manual, (*aims and objectives*);
2. A programmers manual, (*description of programs*);
3. An operators manual, (*description of how to run programs*);
4. A users manual, (*system operating procedures*).

Looking at the lists of documents above it is easy to see why companies need help providing all the necessary documentation. The International Organisation for Standardisation, (ISO), has developed a procedure that includes document creation and control from a quality perspective. The ISO procedures are detailed in section 2.16.1.

2.16.1 The International Organisation for Standardisation Procedures

The International Organisation for Standardisation was established in 1947 and is based in Geneva. The organisation is an attempt by 90 countries to develop common standards by linking their national standards institutions together. The main work of the organisation is to develop international agreements, published as international standards. The organisation reports directly to its membership countries, all of whom pay subscriptions to provide funding, without government intervention, (Rothery 1995).

One set of standards put forward by the International Standards Organisation is the ISO9000 series which was first introduced in 1987. These standards were introduced to help provide accurate and comprehensive control of quality. Rothery, (1995), defines ISO9000 as '*an integrated, global system for optimising the quality effectiveness of a company or organisation, by creating a framework for continuous improvement*'. As part of the quality requirement, documentation is covered in the standard. The documentation topics covered include, (Rothery 1995):

- Exact specifications;
- Precise procedures and instructions;
- Correct descriptions.

The ISO9000 documentation is set at three levels:

1. Quality Manual;
2. Control Documents;
3. Operating Procedures.

The specific documents required can be found in a number of sources. For the automotive components supply chain, to which Ferodo belongs, the quality guidelines followed may be the Chrysler, Ford and General Motors, (1995a), Quality System Requirements QS9000, (which is based on ISO9000) and the VDA Quality System Audit, (VDA 1996). The documents covered are, (VDA 1996):

- Documents on contract review;
- Specifications;
- Drawings;
- Formulations;
- Standards, regulations, internal standards;
- Test instructions;
- Test plans, control plans/inspection instructions;
- Work instructions;
- Work procedures;
- Quality plan;
- Quality procedures;
- Quality manual;
- Procedures for quality record;
- Reference samples;
- Testing procedures.

In addition to defining the documentation required to assist with quality assurance the ISO9000 procedures also show how the documentation should be controlled.

In order to control the documents effectively, the VDA text shows that there five mechanisms that should be in place:

1. Procedures and responsibilities for identifying, generating and releasing documents;
2. Procedures for the distribution and update of documents;
3. Procedures for the archive of documents;
4. Procedures for the timely introduction of external documents;
5. Procedures to ensure that outdated documents are not used.

It was mentioned earlier that the international standards are meant to be a guide to developing procedures. The ISO9000 range of procedures must not be interpreted as an exhaustive list. Unfortunately, many companies assume that minimum compliance with ISO9000 guidelines is satisfactory, (Hoyle 1996). Even following the guidelines of the international standards will not guarantee a useable system if the documentation authors do not focus on the end use of the documents, (Tranmer 1996).

One way to help initially define the documentation required is to model the business at a general level. This provides an information model. One technique that could be used is IDEF0, as described in section 2.11.

It must be pointed out that business documentation is not required solely for compliance with international standards. Documentation is required as part of any Systems Analysis and Design exercise as the systems analysts try to gain an understanding of the process. Business Process Re-engineering, (BPR), a modern business development methodology, also requires the use of structured documentation.

The basic concepts of the BPR methodology are to start from first principles when analysing a system in order to address the fundamental problem and not just to treat the symptoms, (Hammer and Champy 1999). By doing this the company can redefine the business aims, objectives and priorities before undertaking investment projects. The reasons for this are to prevent BPR projects being undertaken that are not financially viable, (Hendley 1997).

In order to help identify key investment projects a BPR project will use modelling tools and structured documentation to help define priorities. The modelling tools could be Input - Output diagrams or IDEF0 diagrams as discussed in section 2.11. The documentation required could cover, (Hendley 1997):

- Business objectives;
- Documentary information flows;
- Definition of requirements from new systems.

This section has illustrated the variety of reasons for creating documents within a business. What has not been shown is how to actually create documentation to meet these requirements. This can be done in one of two ways:

1. The traditional, paper based approach;
2. The electronic approach.

The next section, section 2.16.2, will show the traditional approach to documentation, this will be followed in section 2.16.3 by an explanation of the problems faced by authors and readers of documentation created in this way.

2.16.2 Development of Documentation

Paper based technical documentation has existed in the form of books for approximately 800 years, (Bradford 1988). A number of techniques for document control were developed by the early scribes who created these books. It was not until the development of the printing press that a standard format for documentation was developed from these techniques. The format, which is still in use today, includes the following structuring techniques, (Debs 1988):

- Alphabetisation;
- Annotation;
- Cross-referencing;
- Indices;
- Section titles;
- Title pages.

The technology of paper based information publishing has not developed significantly since then, (the next major development in documentation was electronic publishing media, to be discussed in section 2.17).

Before the advent of electronic publishing media, attention was focused on ensuring that the intended audience type and content of books and other paper based media better met the requirements of the reader, (Charney *et al* 1988). It would appear that there are two potential audiences for technical documents, (Debs 1988):

1. Informational, (*the author is writing the document to provide data to the reader*);
2. Educational, (*the author is writing the document to teach the reader*).

Authors should carefully select the target audience before actually writing a document as they need to know what the audience type will be and also to ensure that the content is set at the correct level, (Charney *et al* 1988). This should be done so that the document can be oriented towards the tasks the users will be expected to achieve and will be understood by those users, (Ramey 1988). One way to help do this is to involve the potential users in the design of the documents, (Knowles 1995).

These documents are usually created using a standard document creation methodology. This methodology has been defined with nine stages as follows, (Brockmann 1990):

1. Develop the document specifications;
2. Prototype the document;
3. Review the document;
4. Write the document;
5. Field test the document;
6. Edit the document;
7. Publish the document;
8. Review the published document;
9. Maintain the document.

Following on from this methodology it can be seen that Brockmann is advocating that the users of a document should be involved in the first three initial design stages, before being invited to field test a document. One other benefit of having users help design documents is that users expect documents to look familiar and will help to ensure this. Defining a departmental or corporate documentation structure is important, as people may react badly when they encounter something unknown, (Debs 1988).

There are two conflicting theories about the content of documents which will be looked at in more detail, (Charney *et al* 1988):

1. The minimalist approach;
2. The elaboration approach.

2.16.2.1 The Minimalist Approach to Documentation

The minimalist approach to documentation is based on the assumption that document readers want to quickly pick up the information and data they require in order to get on with 'doing the job', (this is called task oriented learning), (Charney *et al* 1988). There are a number of rules to help authors create these documents, (Carroll *et al* 1988):

- Focus on real tasks and activities;
- Slash verbiage, (*only include text that is absolutely necessary*);
- Support error recognition and error recovery, (*users will make mistakes, help them learn how to correct these rather than covering everything in order to try to stop mistakes*);
- Guided exploration, (*this can be done by using cards with bullet points rather than textural documents*).

Adult learners have particular documentation requirements, (Brockmann 1990):

- They are impatient and want to learn quickly to be able to get on with a task;
- They rarely read a full manual, they skip to the parts of direct interest;
- They learn best by trial and error, (*learning from mistakes*);
- They are better motivated when they start their own exploration of a subject;
- They are discouraged by large manuals, (*the size is daunting*).

The minimalist approach to documentation aims to address these issues by streamlining documents to provide only the necessary information in small pieces. This approach does have three major drawbacks, (Brockmann 1990):

1. Independent learning can result in gaps appearing in knowledge;
2. The reader needs to be self motivated;
3. It is so different to the normal documentation style that it is difficult to write.

2.16.2.2 The Elaboration Approach to Documentation

The minimalist approach targets detailed learning of tasks. The elaboration approach is aimed at readers who need a broader understanding of a subject, (Charney *et al* 1988).

As the name suggests, authors using the elaboration approach need to fully explain every detail of the work being described and give examples of all the topics covered. By reading this work, the reader may not gain the same task oriented skills as the minimalist documentation reader but they will gain a full understanding of what should be done.

Using these definitions it can be seen that PhD theses follow the elaboration approach, whereas industrial reports to senior management may follow a more minimalist approach. Similarly textbooks aim to give students an understanding of the subject and so are elaborate whereas process operating instructions are task oriented.

In order to provide complete documentation a document author must consider both reader types, task oriented and general learning. In order to provide for both, the author has to create two complete sets of documentation, one for each reader type.

These approaches to documentation also agree with the definition of document types given by Brockmann, (1990). Brockmann defines 4 documents:

1. Reference documents, (*possibly created using the elaboration approach*);
2. Tutorial documents, (*possibly created using the minimalist approach*);
3. Internal documents, (*for example internal training material, minutes and memos*);
4. External documents, (*for example software user guides, warranty documents*).

Using these definitions it can be seen that Internal and External documents will be made up of a combination of Reference and Tutorial documents.

To summarise the work explained in this section, there are four questions that document authors should ask themselves before publishing documents, (Kliem 1984):

1. Is the document written in a logical way?
2. Is the document functional so that the reader can use it to assist with tasks?
3. Is the document clear and easy to follow?
4. Is the document concise and to the point rather than padded out?

With these questions in mind the author can create documents suited exactly to a range of possible reader types by creating a mix of elaborate and minimalist documentation for internal and external use. The work can be published in a paper based format following the structural guidelines explained at the start of this section.

Unfortunately there are a number of problems that have been found with the paper based format. These problems will be explained next, in section 2.16.3.

2.16.3 Problems with Paper Documentation

There are a number of problems with paper based documentation. This section will explain some of the problems found in the literature.

The problems with paper based documentation are, (Kaarela *et al* 1995):

- Poor accessibility;
- Difficulty with maintenance;
- Cost of update;
- Insufficient information content.

These can all be explained as follows.

Poor Accessibility

With paper based documentation and no central library to co-ordinate the documentation it becomes increasingly difficult to find documents of interest. Often it takes so long to find a document that it is out of date before being identified and retrieved, (Sirisawat and Duffill 1997, Knowles 1995).

Difficulty with Maintenance

Many documents change frequently, (Knowles 1995). A result of this is that it is possible for printed manuals to be obsolete shortly after they are released, (Rubens 1988). This will result in a situation where document authors are constantly trying to catch up with current versions of documents and any control over the version in use is lost, (Rousseau *et al* 1994).

Cost of Update

In the case of large manuals and procedures there are many documents involved. An update can mean expensive printing costs as the documents must be replaced, (Sirisawat and Duffill 1997). This problem is made far worse if there is a central documentation department servicing a corporation with separate sites as there will be freight transport costs as well. One issue not covered in the literature found that should also be of concern is the environmental impact of constantly printing documents for distribution.

Insufficient Information Content

The previous section, section 2.16.2, discussed the content of documentation. In this case insufficient content should be read as 'not meeting the requirements of the target audience'. Many documents in use in organisations are verbose. The authors of these documents are trying to put too much into one document, (Tranmer 1996). A symptom of an incorrectly, (or badly), written document is that the documents do not get used and this has a knock on effect as the same mistakes are repeated as no-one has verified the documentation, (Mirel 1988).

It is possible to find other examples of problems caused by ineffective documentation. For example Reynolds and Wainwright, (1997), present the results of a survey that amongst other things showed that many individuals did not know the downstream effect of the information they created. If there was an easy mechanism for those individuals to read about the flow of work within an organisation they may be better able to understand the effect and value of information within the company.

Electronic, paperless, publishing tools can be used to help address some of these issues. Section 2.17 will explain the electronic documentation systems available.

2.17 The Electronic Approach to System Documentation

Another result of the rapid development in computer and software technology explained in section 2.9 is that it is possible to use electronic methods for document creation and distribution, (Bulkeley 1996). The simplest form of the electronic approach is a floppy disk. If a word processed document is stored on this disk and the disk given to a colleague then the document can be read by the colleague on their computer. If, however, the document is stored on a shared network disk drive then the document could be instantaneously accessed by any number of people. Documents stored in this way are considered to be stored ‘on-line’. In a situation where there are a large number of documents stored for retrieval a ‘document management’ system is necessary to help find the documents, these systems will be discussed later, in sections 2.17.1 and 2.17.2. Electronic publishing media address many of the problems found with paper documents which were discussed in the previous section, section 2.16.3, as follows:

Accessibility

Electronic documents are stored within a computer or on a network. With an appropriate retrieval program, (which could be part of an electronic documentation system), these documents can be quickly accessed and distributed, (Rubens 1988). By using a network, the need for centralised documentation is removed and documentation control can be passed to the users, (Edwards and Gibson 1997).

Maintenance

On-line documents can be updated and distributed without the need for lengthy printing and distribution. The process is simply to replace the original document with an updated copy which can then be accessed and read by any number of readers. By reducing paper copies in this way it is possible to work towards the ‘paperless office’, (Baynton 1996).

Information Content

There are two major advantages of electronic publishing when it comes to the allowable content. These advantages are, (Brockmann 1990, Buyers 1984):

1. There are no limits on document size or the number of documents allowed other than disk space, (*in practice, with modern computers, disk space is not a limiting factor*);
2. Graphics, sound, animation and video can be linked into on-line documentation to make it more user-friendly, for example by representing a procedure as a flow chart.

When reading a book it is necessary to adopt a sequential structure to the documentation, starting with the introduction and following through to a conclusion. At the end of a book there is usually an index of key words within the text to help users find sections of particular importance. Electronic documents do not have to follow a sequential structure. There are actually four established structures in use, (Frisse 1988),

1. Random, (*documents have no particular ordering system*);
2. Sequential, (*documents are structured like a book*);
3. Structured, (*documents are arranged into a hierarchy of sequential groupings*);
4. Hierarchical, (*documents are hierarchically structured - like a family tree*).

Frisse explains that in a complex environment documents need to be hierarchically structured to enable the systems managers and users to better understand the way the documents relate to each other.

There are two ways to hierarchically structure documentation:

1. Hierarchical document storage for easy document updates, (Rosman *et al* 1996);
2. Hierarchically organised interface for ease of use by the users, (Brockmann 1990).

It is important to differentiate between these two as there is no reason why the user interface and the document storage hierarchies need to match. The technologies that enable this difference are navigation tools, (Brockmann 1990). With a navigation tool it is possible for the documentation author to create an index for each type of reader. In the case of the documentation undertaken at Ferodo, (see section 3.8), there were three identified readers; readers interested in learning about the flow of items through a product line, readers interested in learning the tasks involved in doing a person's job and readers interested in learning the physical actions to take to do a particular task.

To illustrate how documentation may be developed for these users it is worth using a simple example. For the sake of this example it is assumed that there are five documents that explain the operation of a factory broken down by individual as follows:

1. Purchasing, (*of interest to employee 1*);
2. Planning, (*of interest to employee 1*);
3. Manufacturing, (*of interest to employees 2-10*);
4. Despatching, (*of interest to employee 11*);
5. Accounting, (*of interest to employee 1*).

In the case of the three interested readers the index they require would be as follows:

Readers interested in learning about the flow of items through a product line;

These readers would require an index to guide them through documents 1 to 5 in turn.

Readers interested in learning the tasks involved in doing a person's job

These readers would require an index to guide them through documents 1, 2 and 5 for employee 1, then documents 3 and 4 for the rest of the employees.

<u>Readers interested in learning the physical actions to take to do a particular task</u>

These readers would require an index that guided them to each of the tasks performed by all of the employees.

Any navigation system would greet the user with a simple selection screen asking them which type of reader they are. The reader would select the option, (1-3), that is most appropriate and the system would display the index they require. If their needs change they can simply go back and select another index.

This example is very simple and would not actually require use of these indexes. In an environment where there are many documents, multiple indexes can yield significant benefits in terms of ease of navigation and reduced reader confusion.

Another attribute of electronic systems is that they can search for documents based on user defined key words or key phrases, (Brockmann 1990). These 'keywords' or 'keyphrases' can be either put in a list and associated with each document by the author or the system can be instructed to search the content of every document to find the keywords or keyphrases.

One of the claimed benefits of electronic storage was that documents are allowed to be extremely large. A survey by Brockmann, (1990), has shown that users prefer smaller document sizes when using on-line documents, in same way as for paper based documents. This is where another difference between paper and electronic documents needs to be understood.

A paper based document exists as one physical entity. An electronic document can be split up into a number of on-line pages linked together by use of indexes as above.

In this way it is possible to create extremely large documents containing both the minimalist and the elaboration approaches to documentation. These documents can be split up into a number of pages or modules and indexes created so that the users are pointed to the pages of particular interest.

There is limited advice available on how to actually create the documentation within an electronic documentation system as much of the published work focuses on the hardware and software development rather than the use of the system. Desborough, (1996), advises systems implementers to implement electronic documentation in stages starting with small documents or with existing procedures. By doing this, the implementers can gain an understanding of the operation of the system and develop better working models and practices before moving on.

Desborough also warns that documentation systems are usually developed by technical staff, managed by the documentation authors. He explains that the drawback of this approach is that technical staff can end up creating highly complex systems with all the possible, (potentially redundant), functionality simply because they can. This is a very broad generalisation that based on the author's observations of technical staff in Ferodo is not always true.

Desborough also advocates the use of implementation teams with representatives from the authors, the technical staff and systems analysts in an attempt to co-ordinate the activities towards a better system.

This section has explained the need for electronic documentation and has explained the concepts behind the use of electronic documentation systems. There are two types of electronic documentation systems available, purchased document management systems, (including groupware) and internally developed Intranets. Both of these systems will be reviewed in sections 2.17.1 and 2.17.2. After these sections section 2.17.3 will examine the problems faced by users of these two systems.

2.17.1 Document Management Systems

There are two types of purchased document management systems that will be looked at in this section, Document Management Systems and Groupware.

2.17.1.1 Document Management Systems

Document Management Systems have a number of key functions, (Hendley 1997):

1. Document capture, (*automatically converting a paper document into an electronic document through a process called scanning*);
2. Document creation, (*actually keying a document into an editor or word processor*);
3. Document indexing, (*creating keywords and keyphrases to help later retrieval*);
4. Document storage, (*physically storing the electronic documents and backing up*);
5. Document organisation, (*creation of hierarchies for storage and retrieval and for allocation of document access privileges i.e. read only, edit, delete*);
6. Document distribution, (*new or changed documents automatically sent to readers*);
7. Searching, (*keyword and keyphrase searching as well as full text searches*);
8. Document retrieval, (*displaying documents clearly on screen for reading*);
9. Document editing, (*changing document content*);
10. Document printing, (*printing documents*).

In addition to this list there is Workflow Management, (WM). WM can best be thought of as an automated procedure. A description of WM Software can be found in the introduction to document management booklet by Hendley, (1997):

'*A pro-active system for managing a series of tasks defined in one or more procedures. The system ensures the tasks are passed among the appropriate participants in the correct sequence and completed within set times. Participants may be people or other systems*'.

Workflow Management systems provide block diagram flow charts of the procedures to be followed and act as documentation. These systems are rigidly designed to prompt the users for necessary actions when required. The main advantage of these systems is that they do not allow deviation from the procedure, (this could also be considered to be a disadvantage). With this rigid procedure, once a user has learned the format of the system they can perform any task as the system will always tell them what to do next.

WM systems can be considered to be part of Document Management Systems because the output from some procedures could be documents which then need to be distributed and managed.

Experience of software suppliers has shown that these packages can be bought as three modules to enable users to better meet requirements. The usual modules are:

1. Document capture, (*scanning software*);
2. Document control, (*distribution, editing, printing, retrieval and storage*);
3. Workflow Management.

Document Management Systems are still very new. Recent studies, however, have shown the following advantages of using these systems, (Rosman *et al* 1996):

- Fewer staff are needed to manage and transport documents;
- Electronic document databases can be easily sold to generate income;
- Guaranteed access to documents when needed;
- Immediate access to up-to-date information as soon as it is published;
- Reduced document creation and distribution lead-time;
- Savings in physical storage space;
- Reduced photocopying costs, (*as most copies are made for distribution*);
- With backups of data, valuable documents can be protected against damage;
- Enabling the introduction of Workflow Management.

2.17.1.2 Groupware Systems

A good definition of groupware can also be found from Hendley, (1997):

'*Groupware or workgroup computing packages are specifically designed to support process-oriented ways of working with remote teams and work groups. This software is designed to help groups of workers collaborate*'.

Groupware, therefore, is an information sharing system. Designed as a document database, Groupware systems, such as Lotus Notes, (Greek 1996), are very good at document storage, organisation and retrieval. For this reason Groupware packages are closely linked to Document Management Systems. One feature of Groupware systems is document co-ordination for mobile, (laptop computer), users.

Groupware systems allow mobile users to download databases and documents from the central databases when on-site and take them when they travel, or to copy them across a phone line. In this way group members can be physically separated and still share ideas and discussions.

Typical Groupware functions are, (Hendley 1997):

- Messaging, (*electronic mail*);
- Scheduling / calendar support;
- Group discussion databases;
- Group document management and distribution;
- Electronic forms, (*such as an accident report form or an expenses claim form*).

Groupware packages are very sophisticated pieces of software that are difficult to implement, (Eros 1996). Once implemented, however, it is possible to use Groupware systems to improve competitiveness by ensuring that new processes and information are immediately available to members of a group, (King 1995). For example a salesperson requiring up-to-date information about product availability needs to only dial in and connect to the central database and they will immediately have current product information including notification of changes that have been made to documents.

One of the main strengths of Groupware and Document Management packages is their document storage functionality, (Tabor *et al* 1997). These packages use a series of user defined document databases to store documents. It is these databases that allow the definition of user access, (security), options. It is also this structure that allows whole databases to be selected and copied to transfer current documentation through a group.

2.17.2 Documentation Intranets

The simplicity of the HTML language discussed in section 2.15.3 has led to the development of a new corporate communication system. This system is called an Intranet. Simply put an Intranet is '*a company based communications network using Internet formats*', (Greek 1997). If the Internet can be considered to be a global, open access network of computers an Intranet can be considered as a company specific, private network of computers with a limited target audience and limited external access, (LaLonde and Pugh 1997, McCarthy 1996).

In 1996 there were 30 million Internet users world-wide and this was forecasted to rise to 200 million by the millennium. Despite this amazing growth in the Internet, research has been published that shows that in America Intranets are developing 25% faster than the Internet, (Baynton 1996). It has also been shown that 66% of American companies already use or are planning to use an Intranet, (Tabor *et al* 1997) and that some of these Intranets have as many as 250,000 documents stored on them, (Netscape Communication Corp 1996). Grossman, (1998), warns that British companies need to develop Intranets themselves in order to be able to compete. It would perhaps have been better for this warning to suggest that British companies use electronic publishing media rather than specifically Intranets, however, the message is still clear enough.

It is the simplicity of Intranets that would appear to appeal most to document authors. It has already been mentioned that there are two sides to the use of the internet, authoring and browsing. These tasks can be performed with separate dedicated software tools.

With these two tools the document authors can create documents with the authoring tools, which are easy to use and the readers only need to learn how to use the browsing tools, which are also very easy to use, (Eros 1996). These tools are designed to work under a graphical interface, (for example Microsoft Windows), using a mouse to 'point and click' on-screen at the hypertext links that are to be followed. The look and feel of these 'browsers' are similar to other systems commonly running on PC's, (Hogan 1996). Other benefits of Intranets are that the most popular browsers run on many operating systems and can be acquired for minimal cost or for free, (Eros 1996).

According to Knowles, (1995), the hardware and software elements of an Intranet are only one of the factors necessary for success. Perhaps a more important consideration is that this technology must be carefully applied to ensure a satisfactory system.

There are three main uses of Intranets, (Tabor *et al* 1997):

1. Internal bulletin boards, (*distribution of corporate news and discussion groups*);
2. Information distribution, (*documentation and graphical files*);
3. Support for business functions, (*developing communication between departments*).

As already mentioned in point 2 above the documentation that can be distributed using an Intranet does not need to be textural, it can be graphical also. Tian *et al*, (1997), show four types of document that can be distributed on the Internet or an Intranet:

1. Simple textural information;
2. Multimedia documents, (*documents with graphical, audio and animated content*);
3. Applications that require reader interaction, (*i.e. online forms*);
4. Applications that deliver software code, (*downloading software*).

It is essential that Intranets are implemented to meet the current and future requirements of the authors and users. One example of this is graphics files. Anyone who has used the Internet will be used to the waiting involved in downloading pages, (called 'webpages', which are accessed in a particular application by following hypertext links from the documentation author's 'homepage'). The delay comes from the physical transmission capacity of the equipment being used. One way to speed up this information download is to instruct the browser software not to download graphics files and just to download text. There is a lesson in this for documentation authors. Graphics should be used where necessary and not simply because it is possible.

Many authors have written about Intranet implementation. Knowles, (1995), suggests that Intranets should be implemented as a company wide system but that a modular approach should be used to enable a learning process to be followed. Implementers are also advised that in anything other than a simple implementation an established design methodology should be followed for two reasons, (Rosman *et al* 1996):

1. Experience gained from previous implementations can be used to develop tested and proven methods to assist with the implementation;
2. A design methodology will ensure a standard Intranet format is used across the organisation reducing the chance of user confusion when faced with different looking systems.

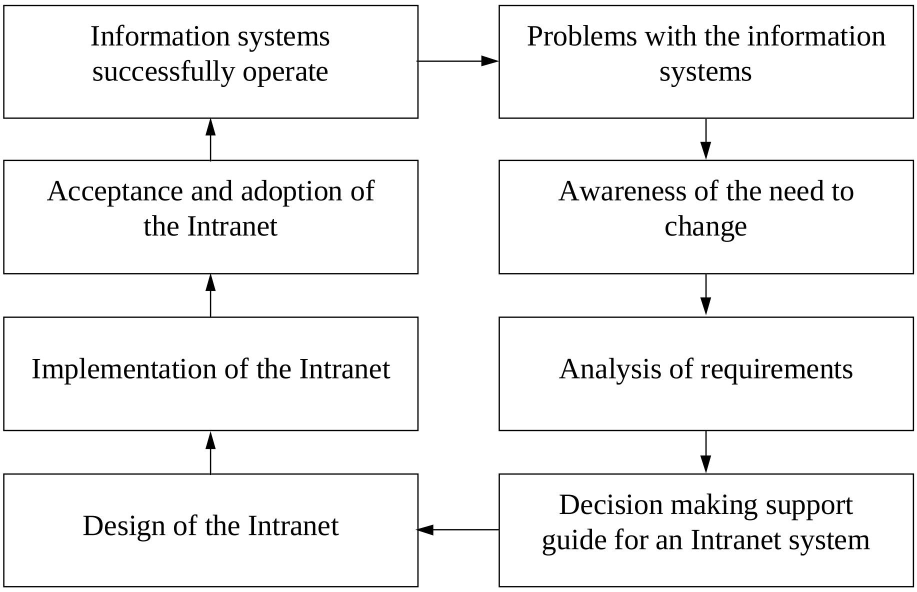
Rosman *et al*, (1996), go on to suggest that much of the equipment and software for an Intranet, (or any other documentation system), will be purchased rather than being developed in-house. Based on this suggestion they illustrate an implementation path.

The implementation path suggested is, (Rosman *et al* 1996):

1. Situation analysis and definition study;
2. Preselection of software packages;
3. Invitation to tender a quote;
4. Selection of package and contract negotiations;
5. Installation, testing and use.

The same path can also be followed for the hardware issues involved in implementing documentation systems. Sirisawat and Duffill, (1997), have proposed a methodology for the software implementation, which they call a decision support tool. The methodology starts in the top right hand corner with the identification of a need. Sirisawat and Duffill have defined a circular methodology, illustrating the need for the continuous development of the systems as the successful completion of a project leads into the definition of a new problem situation. This methodology is illustrated in Figure 24 below, (Sirisawat and Duffill 1997).

Figure 24: Framework for Planning, Implementing and Managing an Intranet



At the start of this section the early development of the Internet was attributed to Tim Berners-Lee. Berners-Lee has published, (on the Internet), a document that attempts to apply rules to the content of Internet and Intranet documents, (Berners-Lee 1998):

- Adopt a hierarchical tree structure after initially consulting the end users;
- Provide internal links to cover all the potential user's requirements;
- A document should be no longer than five pages;
- Provide links to all referenced material;
- Explain technical abbreviations;
- Test the documents before publishing, with real users;
- Give documents status within the company;
- Sign the documents to attribute ownership.

Software tools for documentation would appear to solve many problems found with traditional, paper based, documentation. It has just been shown that care must be taken when actually creating documentation systems over the content and appearance of the documentation. The next section, section 2.17.3, will illustrate other problems found by users and developers of these systems.

2.17.3 Problems with the Electronic Approach to System Documentation

There are a few general problems with electronic documentation systems that will be investigated at the start of this section before looking at specific problems with the two types of documentation system. The first problem, which all Internet users will have encountered, is that links between documents are only valid as long as target documents are not moved or deleted.

If a document is moved or replaced then the link will point to either a different document or nothing at all and there will be no indication of the new location of the document. These links themselves are difficult to create if there are a lot of documents that all need linking to each other, (Agosti *et al* 1996). There is work being done on automating link creation by searching for word patterns within documents, (Kaarela *et al* 1995), though for good results all the documents in the system must contain the word patterns and not different versions of words, (for example a lorry can also be described as a van or a commercial vehicle). Even successful creation of links can be problematic as it is not always easy for users to successfully navigate through documentation without getting lost, (Frisse 1988).

Electronic documents are also so different from paper based documentation that there are hardware and software costs as well as extensive staff training costs to be borne, (Rosman *et al* 1996). Adequate resources also need to be allocated to the systems management department. Another potential problem, described by Barua and Ravindran, (1996), is that whilst technology allows effective information sharing it can stop this communication if it is not properly managed.

2.17.4 Specific Problems with Document Management Systems

Document Management Systems are packaged systems that are configured to meet the needs of a particular business. This can result in a system that is too prescriptive and therefore, cannot adapt as the needs of the business develop, (Hoyle 1996). Care needs to be taken when selecting packages to define current and future requirements.

These systems are also very difficult to implement, due to their sophistication and complexity, (Eros 1996). Finally, experience gained from a review of this type of software by the author shows that these systems are extremely expensive. A verbal quote received by the author came to £500,000 to cover all Ferodo's employees despite most being readers not authors. Intranet browsing software is free. It is for these reasons that Intranets are such a threat to Document Management Systems, (Inman 1996).

2.17.5 Specific Problems with Intranets

There are two problems with Intranet technology. Firstly there is no built in mechanism for structured file storage. The only methods available are the use of hierarchical directories or structured file names. As a result of this it can become difficult to locate physical files as one of the strengths of Intranets is that they are 'unstructured'. In addition to this, Intranets have no mechanism to notify users of changes made to documents. Document Management Systems can be considered to be 'Push' type systems, where readers are notified of changes to documents of interest to them. Intranets are 'Pull' type systems. Readers of Intranets do not get notified of changes, they have to find changes themselves, (Greek 1996). As a result of this lack of feed-back the document authors have no guarantee that their changes have been read.

It would appear that the solution would require a mechanism to combine the structure of a Document Management system with the unstructured document access provided by an Intranet. Work undertaken by the author in Ferodo has attempted to meet these requirements using a version of Lotus Notes called Lotus Domino which allows documents stored in its database to be converted into HTML files, (Borysowich 1996). This work is described in section 3.10.

2.18 The Ferodo OE Manufacturing Planning and Control System

As mentioned in section 1.3, Ferodo OE implemented an integrated MRPII package called MFG/PRO in 1994. This was done by first assigning an implementation team which reported to an overall steering committee. The team chose to use a phased implementation, as introduced in section 2.3 and decided upon two rules to govern the implementation effort:

1. A core system was to be defined and implemented to cover all of the essential business systems. A number of low-priority sub-systems were to be left running as standalone applications to be dealt with at a later date as resources became available.

2. The core system was to be designed and used without making modifications to the operation of the database package. Anything that was not covered by operation of the database was to be either left as a standalone application or created as a separate database integrated with the core.

The following sub-systems became priorities in later years:

1. A system to record and analyse internal non-conformances;
2. A system to display manufacturing and process control data;
3. A system to record and analyse, for statistical process control purposes, the physical property test results from the factory.

In addition to the three systems listed above, the author spent time looking at the issue of Capacity Planning and in particular FCS with a view to integrating a package with the core system to assist with planning the factory.

Recently Ferodo have also considered controlling part of the factory using a Kanban or Pull system rather than straightforward MRPII wherever there is a repetitive manufacturing environment and have started a Lean Manufacturing campaign to try to further integrate the concepts of JIT and MRP.

This chapter has provided a theoretical foundation for the research project. It has explained three of the main Manufacturing Planning and Control systems in use today, has discussed some of the problems facing these systems and has shown some of the solutions that can be adopted to address these problems. The chapter has also shown that whilst all three Manufacturing Planning and Control systems are different in application they all have the same data storage requirements and can use a similar database structure to do this.

This chapter has also discussed the need for structured documentation and has presented a number of different standards and techniques that can be used to help control the documentation process. The chapter has also shown how electronic publishing can be used to simplify the documentation process, though care needs to be taken to structure the content as this structure is not provided by the approaches reviewed.

Chapter 3 will explain the techniques used by the author to develop the Ferodo OE systems. Experience gained through use of these techniques will be used to develop a generalised methodology for MRPII development. Following on from this section 3.8 will explain documentation work undertaken by the author and the design of an 'Intranet' with a feedback loop developed to better distribute and control documentation. A methodology for MRPII structured system documentation will be derived from the author's experience of designing the MRPII documentation Intranet.

Chapter 3. Structured MPC System Development

3.1 Introduction

Chapter 1 introduced the research objectives and the research framework. The research objectives required the researcher to gain an understanding of the subject area, from literature review and by analysing some Manufacturing Planning and Control, (MPC), systems in practice. The framework was shown to be conceptual, that is that the basis and boundaries of the research would be determined whilst conducting the research.

Chapter 2 presented a literature review that investigated the development of MPC systems, the development of Information Technology, Systems Analysis and Design and Modelling techniques and the development of structured approaches to documentation. This constitutes the theoretical basis of the research, helping to satisfy the objectives by educating the researcher and helping to guide the research by identifying problem areas.

This chapter will complete the education phase of the research by discussing a number of case study visits to companies with a range of different systems, (section 3.2), by analysing an early systems development exercise at the case study company and by defining the ‘satellite systems’ approach to systems development, (section 3.3 and the start of section 3.4). It will be shown that the case study company can be considered to be a representative test environment.

The chapter will continue by detailing some further systems development work undertaken by the author, (section 3.4). Development of these systems helped the author develop a clear definition of the ‘satellite systems’ approach and was the basis of the development of the systems development methodology.

Using the knowledge gained through the literature review and the systems development projects at the collaborating company the author developed the systems development methodology. This chapter will present the methodology, accompanied by some decision support tools; Decision Support Questions and Indicative Ratios, (section 3.5).

The general applicability of the methodology will then be discussed, (section 3.6), building on the findings of the literature review and the case study findings. This will be followed by a discussion of some problems found whilst applying the 'satellite systems' approach within the case study company, (section 3.7).

This chapter will continue by investigating structured documentation. Building upon the theoretical foundation provided in Chapter 2 it will start by highlighting the need for structured documentation, (section 3.8). In keeping with the research objectives the author produced system documentation at the case study company. A paper-based approach to document distribution was adopted, (section 3.9).

The author developed an Intranet for electronic publication of the documentation, (section 3.10). Whilst this resolved some of the problems faced when creating documentation, such as speed of information transmission, a structured approach was deemed to be necessary. Using the experience gained whilst documenting the existing systems the systems development methodology was enhanced by including a structured approach to documentation, incorporating a template of a manufacturing business and a definition of the levels of documentation that are required, (section 3.11).

The chapter will end by discussing the general applicability of the newly developed methodology, (section 3.12).

3.2 Case Study Companies

In order to develop a fuller understanding of the nature and use of MPC systems a number of companies were worked with, to different degrees, or visited, in addition to the collaborating company. These companies supply, or are associated with, the Federal Mogul Corporation and were either planning to install, installing or already using MPC systems. The companies worked with were, (*these are reviewed below*):

1. **Ferodo Aftermarket Manufacturing Division**, part of Federal Mogul Friction Products, an Aftermarket Friction Material manufacturer who use MFG/PRO;
2. **Bentley Harris** in Rochdale, part of the Federal Mogul General Products Group who use MFG/PRO and were implementing Finite Capacity Scheduling;
3. **Eurobrakes Ltd.** in Birmingham, a raw material supplier, who were reviewing the possibility of using an MRPII package to plan the factory;
4. **Eurofriction Ltd.** in Kilmarnock, part of Federal Mogul Friction Products, an Aftermarket Friction Material manufacturer, implementing MFG/PRO;
5. **Ferodo Railways Division**, part of Federal Mogul Friction Products, a specialist manufacturer of Railway Friction Material, using MFG/PRO;

The companies visited were, (*these are not reviewed below*):

6. **AE Goetze** in Bradford, part of the Federal Mogul Pistons Group, who use an MRPII package and a Finite Capacity Scheduling package;
7. **Farrington Components** in Leyland who implemented OPT.
8. **Leicester Engineering** in Leicester, a raw material supplier who were implementing a Finite Capacity Scheduling package.
9. **Weyburn-Lydmet** in Gloucestershire, part of the Federal Mogul General Products Group, manufacturers of camshafts who were implementing MFG/PRO.

3.2.1 Ferodo Aftermarket Manufacturing Division

Ferodo Railways Division was one of the first T&N companies to implement MFG/PRO. It was chosen as a pilot site for implementation as it is relatively small. According to members of the Ferodo implementation teams, experience gained in the Railways Division was invaluable for later projects in other divisions, (these implementations used separate project teams, each with a member from the Railways team). A result of this multi-team strategy was that the details of the implementations differed, despite regular meetings to share ideas.

The Aftermarket manufacturing environment works on far shorter lead-times and greater product mix than the OE manufacturing environment. A result of this is that the two MRPII implementations had different design constraints imposed. It was decided within the Aftermarket Division that the functionality of the standard system did not meet requirements. There were a number of programming changes made to the standard system to ensure that the requirements were met. For example work order processing programs were modified to streamline work flow by product line. The desired effect was achieved and the implemented systems were tailored to match business need. Documentation was produced, at an overview level and detailed operating procedures were passed down between staff during training. Despite having tailored systems the Ferodo Aftermarket division successfully started implementing the quality systems developed by the author in the Ferodo OE Division.

When a recent upgrade was made to the Aftermarket database it was necessary to find and update all the special modifications made to the operating programs of the package. This task was achieved with significant effort despite Ferodo being large enough to support a central team of software programmers to assist with this task.

3.2.2 Bentley Harris

Based in Rochdale, Lancashire, manufacturing insulating sheaths for electrical wiring looms, Bentley Harris implemented an MRPII package in the early 1990's. The manufacturing process involved producing rolls of assembled material which was then slit to customer order to produce the finished components. It was decided in 1997 that the infinite capacity scheduling algorithms of an MRPII package could not produce an effective, detailed plan for production of the finished components and that implementation of a Finite Capacity Scheduling (FCS), package was necessary.

The FCS package was implemented alongside the MRPII package with an interface to translate data between the two packages. It quickly became obvious that in order to effectively run the FCS package, a high level of data accuracy was necessary within the MRPII package. In addition to this it was found that it was necessary to review working practices to ensure that the data was kept accurate.

There were a number of problems that the implementation effort faced, in particular there was an attempt made to work with a customised interface between the two packages. This customised interface caused a number of compatibility problems and eventually had to be removed and the standard interface used.

It was seen that it is necessary to have a stable basic MRPII implementation before considering the expansion of the system with auxiliary packages. It was also noted that Bentley Harris were another company that had attempted to modify a standard package and encountered problems as a result of doing so.

3.2.3 Eurobrakes Ltd

Eurobrakes Ltd were a supplier of metal plates to the Ferodo Aftermarket Division. There was concern over the performance of this company and it was decided that a review of their MPC systems was necessary. It was discovered that the MPC systems within Eurobrakes were predominantly manual, that they had difficulty maintaining visibility over work in progress and that it was also difficult to reconcile product pending despatch with end customers. In addition to this they were operating with a significant backlog of orders in arrears.

A consultant had suggested that the company adopt a cellular manufacturing approach and as a result Eurobrakes were adopting this approach. It was hoped that the use of cells would streamline material flow, reducing work in progress and therefore increasing visibility of work in progress. It was obvious to Eurobrakes that this would not completely address the problems with factory scheduling and customer order despatch.

This may appear to be a situation where implementation of a computerised MPC system would add benefit. The small scale of the business and the price of these systems, however, precluded that option.

After detailed analysis it was suggested that Eurobrakes re-schedule their arrears forward and use manual work order control, either using pre-printed stationary or large white boards in the factory to control production and allocate finished stock to customer orders. It was determined that any systems development methodology involving purchased computer systems would only be applicable to those companies that can afford to implement large MPC systems and that in some instances a change in operating procedures would be sufficient.

3.2.4 Eurofriction Ltd

Eurofriction were a sister company within the Friction Products group, making friction products for the Aftermarket market. They were just starting to implement MFG/PRO in line with the rest of the Friction Products group. The company, being of a far smaller scale than Ferodo, could not dedicate a project team and as a result the team was 'part time'.

It was decided that the implementation would attempt to use the standard functionality of MFG/PRO but would try to maintain consistency with the Ferodo Aftermarket Division systems.

On a long term basis Eurofriction were aware that there would be certain issues that would need addressing and decided to emulate the core and sub-system approach followed by the Ferodo OE Division. In particular, interest was expressed in the quality systems developed by the author in the Ferodo OE Division.

In addition to using a modular approach to implementing MFG/PRO, it was apparent that a documentation system would be of interest to Eurofriction and that a structured approach to systems development would add benefit. On a short term basis it was decided that the Ferodo OE systems would be investigated when the Eurofriction implementation was complete and stabilised.

The Eurofriction implementation was another indication that transferable sub-systems would add benefit to a number of companies operating with the same MPC system.

3.2.5 Ferodo Railways Division

It has already been mentioned that Ferodo decided to implement the MRPII package in stages, starting with a pilot implementation. It was recognised that there were three parts of the manufacturing business, OE, Aftermarket and Railways and that all three required separate MRPII databases. The company was divisionalised along these lines as described in section 1.3.

Of the three divisions the Railways division was by far the smallest and was chosen as an ideal candidate for a pilot implementation. A dedicated project team was formed to undertake this pilot implementation.

Once the implementation was finished the members of the project team were split amongst the remaining implementations in order to ensure that the necessary skills were evenly distributed. Whenever an upgrade of the MRPII package is undertaken at Ferodo, the Railways database is always the first to be addressed.

When the quality systems presented in section 3.4 were being developed by the author within the Ferodo OE division the Railways division expressed interest in adopting the systems for their own use. When the systems were analysed it was discovered that the quality systems would easily integrate into the Railways database.

The Ferodo Railways division illustrates the benefits of the use of a pilot implementation. It is also another example of the use of transferable sub-systems designed to operate with a particular MPC system.

3.2.6 Case Study Findings

It has been shown that these companies represented a wide range of organisations, both in the OE and Aftermarket sectors. The larger companies had dedicated systems staff, the smaller ones used existing staff in a multi-function role. There were, however, a number of similarities between all these companies and with the Ferodo OE division:

- They all used Work Order management in some form to control the factory;
- They all made items, in batches, (*therefore, all are 'discrete batch' manufacturers*);
- They all expressed an interest in developing the systems to increase functionality.

It was clear from informal discussions with selected personnel from the companies that they all followed or intended to follow a similar implementation methodology that centred around the choice of MPC package. The typical approach to implementation was to choose a package supplier and then to choose the modules that best suited the business. These modules were then to be implemented, either all at once or one at a time, until they were all in use. Anything not covered by the MPC systems was to be left standalone, (separate from the main MPC systems), apart from the Ferodo Aftermarket Division which modified some of the programming. Inconsistencies arise when trying to decide how to develop the use of the systems once they have been implemented. There are several approaches that can be adopted for this systems development.

The case studies showed that Ferodo was not atypical of manufacturing business. Increasing demand for on time deliveries and more detailed Manufacturing Planning and Control drove all the case study companies towards considering investment in computerised MPC systems and in this respect Ferodo was no different.

It was concluded, therefore, that research undertaken at Ferodo in the area of MRPII system design and development could be assumed to be of general applicability.

Further to this a detailed analysis of the Ferodo MPC System allows conclusions to be reached as to the success of the systems. These conclusions are explained below:

1. The MRPII system used by Ferodo was relatively unmodified, as discussed in section 2.18. Care had been taken to ensure that the system was implemented in an unmodified state. As a result the system could be easily upgraded when applicable;
2. Structured, formal system reviews were undertaken. Members of the original project team met up on regular, (usually monthly), intervals to review the MRPII system and any development work that was necessary, as discussed in section 1.1. It was as a result of this structured review that the PhD research project described in this thesis was proposed;
3. Section 3.9 will describe the documentation exercise undertaken by the author within Ferodo. When conducting these interviews it became apparent that the formal systems, (the procedures that management believed were followed by their staff) and the informal systems, (the procedures that staff actually carried out), matched;
4. During the three years that the author worked at Ferodo undertaking research the only major problems encountered with the operation of the MRPII system were hardware related; disk space, (as the database grew in size) and access speed (as more staff required access to the advanced reporting capability offered by MRPII systems).

This chapter continues by looking at techniques for the development of databases and in particular MRPII databases. It will go on to propose an approach that is based on the definition of a ‘core’ and subsequent ‘satellite’ systems. This ‘satellite systems’ approach will be developed into a methodology for MRPII systems development.

3.3 Early Systems Development at Ferodo

Using the core and sub-systems approach the Quality Analysis system was developed at Ferodo and integrated with the MRPII package used by Ferodo, MFG/PRO. The Quality Analysis system is reviewed in section 3.4.1 below. The author started at Ferodo after the Quality Analysis system had been developed and integrated. A review of the work done on systems development was commenced in order to better define the core and sub-systems approach to MRPII development.

It was clear that the 'core' system described by the earlier authors, (McLeod *et al* 1995 and Harrison *et al* 1995), was a purchased MRPII package and therefore, it was not necessarily a physically centralised system but rather a modular system with a number of sub-systems. It also became evident that the sub-systems being integrated with the MRPII package and the sub-systems comprising the MRPII package were different. For this reason the phrases 'core' and 'satellite' system were defined as below.

1. The core system is the purchased MRPII package which itself can be a centralised or distributed database.
2. Any sub-system being integrated with a core database, using a hybrid approach to integration is a satellite system to the core as a whole, despite being a sub-system of a centralised database or a sub-system of one database within a distributed database.

In order to better understand this 'satellite systems' approach to MRPII systems integration the author integrated two new systems, (the Property Recording and Manufacturing Instructions systems), with the Ferodo MRPII package. The systems are reviewed in section 3.4.2 and section 3.4.3. It was expected that experience of the application of satellite systems would enable the development of the methodology.

3.4 Satellite Systems Integrated at Ferodo

In 1994 Ferodo implemented MFG/PRO in the OE division, (see section 1.4). The implementers decided that the package was to be implemented without modification and would form the basis of what has now been defined as a 'core' system. The implementation philosophy was to attempt to include as much data and information in this core system as was possible. This was to be achieved by using existing modules and by using spare textural fields to store information that did not easily fit into the available modules. This textural information was converted into meaningful information by running reports to extract and analyse the stored data.

Once this core system was in place it was developed by integrating a sub-system, that could not easily be assimilated into the core, by following a hybrid approach to systems development. The development of the core system was explained in a number of publications, (Petty and Harrison 1995, Harrison *et al* 1995) and led to the system illustrated in Figure 3.

The system integrated initially by members of the implementation team was the Quality Analysis system to record and report on internal non-conformances.

The work undertaken by the author within the OE division was to continue to integrate systems into the core system using sub-systems. This experience resulted in better definition of this approach, to be called the 'satellite systems' approach which was developed into a methodology as presented in section 3.5. The code for the systems designed and implemented by the author was written in-house at Ferodo by a team of experienced programmers with close support from the author.

The satellite systems that the author worked on in the OE division at Ferodo were:

Property Recording

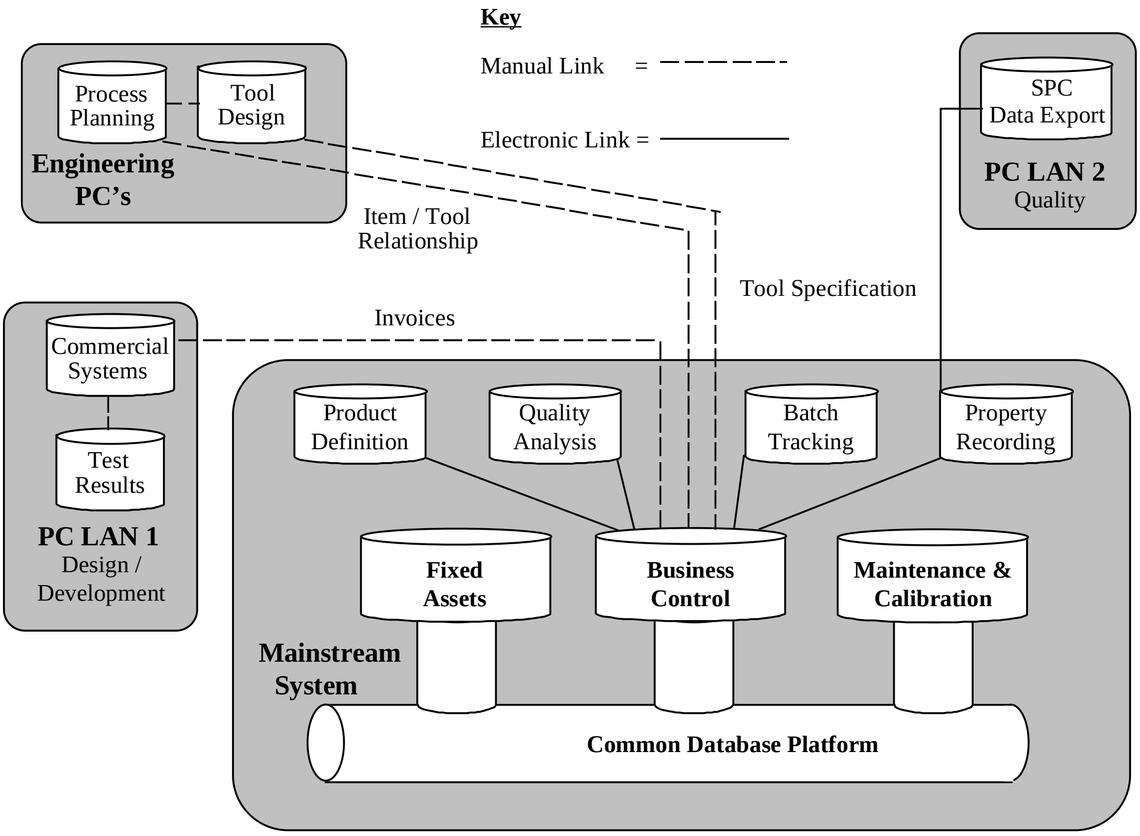
A system to store, analyse and display statistical process control data. (This project took approximately 1 year to complete).

Manufacturing Instructions (Product Definition)

A system to store, display and control the update of product specific data. (This project took approximately 6 months to complete).

Once the new satellite systems were implemented the developed systems structure was as in Figure 25 where the MFG/PRO package is the core and the Product Definition, Quality Analysis and Property Recording systems are satellites.

Figure 25: Developed Ferodo OE Information System



The three systems integrated as satellites will be reviewed in sections 3.4.1, 3.4.2 and 3.4.3, section 3.4.4 will describe the batch tracking system that these systems enabled.

3.4.1 Quality Analysis System

The manufacturing modules of the core system implemented at Ferodo are used to authorise manufacturing staff to make product by issuing Work Order cards for discrete batches of work. When product has been made, there is the possibility that some product does not conform to quality specifications, these products are called Internal Non-Conformances, (INCs). The core system allows entry of the quantity of products that have not met specification so that the stocks can be adjusted accordingly. For example if a planner authorises the manufacture of 1000 parts and 5 are INCs then they can only sell 995. 1000 parts of raw material still need to be issued from stocks for data accuracy, however, as INCs consume stock.

Though the core system allows entry of total numbers of INCs, it does not allow entry of a breakdown of the INCs found, such as cosmetic blemishes or incorrect dimensions. This information was held in standalone databases with duplicated keying required and no automatic data verification mechanism.

It was decided by members of the implementation team that a sub-system should be designed to store and analyse this information and that it should be integrated with the core system to reduce duplication and allow data verification. This sub-system became the first ‘satellite system’ to be integrated with the OE database, (Harrison *et al* 1995).

The new Quality Analysis system consists of two files, a datafile to store collected INC data and a control file that contains the headings with a link between them. The data and headings when combined give meaningful information.

The files are detailed in the appendices as section 6.1. The system performs some calculations on the data entered into these files and looks up data from the core system as follows:

- A running total of the INCs entered in the Quality Analysis system is displayed, excluding all that are not to be included in calculations;
- A running total of all the INCs entered in the Quality Analysis system is displayed;
- The total number of good items entered in the core system is displayed;
- The total number of INCs entered in the core system is displayed;

At the end of the data entry process, the system will check that the total of the INCs entered in the Quality Analysis system, excluding all that are not to be included in calculations, does not differ from the number entered into the core system by more than 1%. If there is a difference it will prompt the user to check and re-enter the data if required.

One other problem found with the core system was that it does not automatically warn if the quantity of product entered when finishing a Work Order does not match the quantity ordered. The Quality Analysis system does this check. The system adds the total number of good items entered into the core system to the total of the INCs entered in the Quality Analysis system, excluding all that are not to be included in calculations. If there is a greater than 25% deviation from the quantity ordered an error message is displayed on screen and the user has the option to re-enter data.

A programmer wrote a program that extracts the data stored in the two datafiles into a downloadable file that can be read into a standard spreadsheet for detailed analysis.

There are two ways that users can finish Work Orders in the core system, (called Work Order completion). Each of the screens used to do this have a spare text field at the bottom for the storage of miscellaneous data. The decision was made to use these fields as gateways to the Quality Analysis system. Once the user has entered the number of good products and the total number of INCs against a Work Order they move to the text field and press function key F2 which activates the Quality Analysis system.

The system already has the Work Order Identification Number, (Work Order ID), as the user was in a Work Order completion screen. Based on this ID number the system identifies the product line from within the core system and looks at the control file entries for that product line to create a suitable data entry screen with the headings listed next to blank fields for data entry. Once data has been entered against all applicable INC types the system performs the calculations and exits back into the original Work Order completion screen if there are no data errors, (as explained above).

The Quality Analysis system, therefore, operates as a self contained system, allowing the user to input data not normally, or easily, stored in the core system and performs analysis on data entered. The system is written in the language of the core and is integrated with the core by use of the Work Order ID number which the system reads automatically from the Work Order completion screen being used to finish off Work Orders from the factory. The system uses this Work Order ID number to interrogate the core system to determine the Item Number, quantity required and the product line of the item. The system uses the product line to then create necessary headings and rules for the data it receives from the user from the user defined control file.

3.4.2 Property Recording System

The work of W. Edwards Deming described in section 2.6 involved teaching Statistical Process Control, (SPC). SPC works by analysing physical tests performed on finished or semi-finished product and comparing those tests with history to determine whether the process that created the product is operating within statistical control. An example will help to explain the basics of SPC. A product specification shows that a particular dimension of a product must be between 1.20 and 1.30 metres and seven separate tests are taken for seven separate Work Orders. The readings taken are as follows:

Test 1 Dimension = 1.23 metres;

Test 2 Dimension = 1.24 metres;

Test 3 Dimension = 1.25 metres;

Test 4 Dimension = 1.26 metres;

Test 5 Dimension = 1.27 metres;

Test 6 Dimension = 1.28 metres;

Test 7 Dimension = 1.29 metres;

Analysis of these rising readings shows that test 8 will probably have a dimension of 1.3 metres and test 9 will be out of specification. In this case the trend is that the readings are rising by 0.01 metres each works order towards the upper specification limit. There are a number of different trend analyses that can be performed on physical test results, (Chrysler, Ford and General Motors 1995b). The results of these analyses can help to predict future quality problems and allow the users to prevent them or to help determine the underlying cause of a problem. The rules and operation of SPC are detailed in the appendices as section 6.2.

The quality function originally appointed a quality systems manager to create and maintain systems to undertake these analyses. Each product line had its own database, written and stored using a purchased package. Each database was developed independently of the others and the end result was that there were a number of different versions of the purchased package used for a number of different databases of different structures. Physical testing data was entered manually by the user onto a PC network as soon as testing was complete. Most of the physical testing areas had their own network workstations to speed up this process. This data was sorted by Item Number and Work Order ID for manufactured items and by Item Number and batch number for purchased items. This ensured fast analysis of data which is important for SPC. The systems all operated independently of each other and the core, (once it was implemented) and so had no mechanism for automatic data verification. These databases became very hard to support and ensuring data integrity was difficult.

These databases were linked, (by use of DOS batch files), to a propriety SPC package called Quality Analyst which performed an analysis on the data entered along with historical data that was also sent to it. The results were displayed on screen. If the trend analyses showed a potential problem an error message was displayed and the user sent a printout of the results to the quality laboratory. Observation of the actual operation of the system by the author showed the following drawbacks:

- Not all the computers were located near testing benches and as a result there was no automatic check to ensure that sufficient information was keyed into the system;
- The users sometimes did not notice the error messages, (which form part of the early detection and correction procedure);
- There was no automatic mechanism to verify that the data entered was as expected.

A drawback of the systems, experienced by the author when designing the new systems, was that due to using an SPC package there was a risk that the users may not develop a detailed understanding of the operation of SPC. It has been argued that quality systems must be developed in-house so that they are fully understood, (Hoyle 1996).

The author applied the satellite systems approach to help design a new system that was to have the following characteristics and functions:

- Data verification with the core system;
- Absolute specification tests to be performed with user defined limits;
- Trend analyses to be performed with user defined analysis criteria;
- Automatic notification of trend analysis warnings;
- Action lists of Work and Purchase Orders that have not been fully tested;
- A transportable system for use with other MFG/PRO databases;
- A self contained system to be integrated with the core through key fields.

For manufactured items the key field chosen was the Work Order ID field as the tests were carried out on Work Order defined batches of product. For purchased items the Purchase Order Identification Number, (Purchase Order ID), does not uniquely identify product as multiple receipts can be made against a Purchase Order ID. As the core system generates a unique transaction number whenever an item is received into stock, it was decided to use this as the key field for each batch of purchased product.

The new Property Recording system consists of four files and a number of reports. The four files are a datafile to store test results and three control files to store static information. The files and reports are detailed in the appendices as section 6.3.

The Property Recording system is a self contained system written in the language of the core, stored within the core DMS and operating as part of the core. All the logic required to run this system is contained within the system itself and no modification is required to the core system. The system is linked to the Work Order ID number and Item Number of manufactured items and the Purchase Order ID, Item Number and Transaction Number of purchased items.

The users input data into the system, identifying the batch of product being tested, which is then compared against data stored in the core system to verify that there are actually items requiring testing with valid identification numbers. Once the user has identified the item in question the system looks up the standard tests that are performed against the item and creates an appropriate data entry screen.

The user keys test data into the system. This test data is analysed against absolute specification limits for the item in question and any violations of these limits are brought to the user's attention so that they can decide upon the appropriate action.

Once the absolute specification limit test is passed the system performs statistical process control trend analyses on the data, based on pre-defined statistical process control trend analysis rules. Any violations of these rules are displayed on screen and written into a rules violation file to be investigated by the quality laboratory.

As the system is totally self contained it can be transported to any MFG/PRO core database with discrete batch manufacturing using Work Orders and Purchase Orders.

3.4.3 Manufacturing Instructions System

The Ferodo Manufacturing Instructions were paper based documents that provided all the information necessary for the manufacture of product. The information was contained in a series of paper documents which were created using a word processor and stored in a series of lockable books. Each section in the factory had a lockable book of relevant Manufacturing Instructions which were held in approved and usually secure locations around the site.

There were two changes that could be made to the Manufacturing Instructions:

1. An existing Manufacturing Instruction could be amended with new data;
2. A new Manufacturing Instruction could be added.

These changes only occurred if the change was to be permanent. Whenever a change or addition was made to the Manufacturing Instructions, it required distribution of corrected documents to each approved location holding a copy of the particular instruction.

When documents were to be amended a Change Control Form was sent to the manufacturing standards department by the amendment originator to notify intention to change. The change was carried out by the amendment originator and a Change Note was filled in as a cross-reference.

The Change Note and the amended document were then sent to the Manufacturing Standards Department where the Change Note and the amended document were compared against the Change Control Form. If the changes were satisfactory, the Manufacturing Instruction was replaced by the new document and a copy distributed to relevant locations.

The procedure was simpler for new documents. A copy was taken of the existing Manufacturing Instruction that most closely matched the new document by the new document originator. This copy was modified to show the process being documented and sent to the manufacturing standards department for distribution to the relevant locations.

The documentation system highlighted above proved ineffective because:

- The authoring procedure was lengthy, resulting in difficulty in establishing which version of a document was current if another change became necessary whilst authoring was in process;
- The information presented to the users in the factory may have been dispersed throughout a 10 page document. As a result they sometimes copied information down to condense it. This meant that there was no longer any control on version numbers as these copies were not registered.

It was observed that the Manufacturing Instructions comprised of two discrete entities:

1. Purely text based process descriptions, serving a range of items, which rarely changed;
2. Purely numerical data specific to an item which frequently changed.

Based on this observation a new model of a Manufacturing Instruction was developed. The model comprised of a database of item specific data and a series of text files describing the processes used in the factory and the origins of the data in the database.

Users of the system would read the text based instructions, (Process Instructions) and interrogate the item data database, (Product Instructions), to find the relevant data. Only the most recent version of the data would be available in the Product Instructions database.

A tabular, structured database is extremely easy to maintain and backup. Amendments and additions, which are usually numerical in content, would be simple. Infrequent text based amendments would be more complex to do but only one document would need changing.

It was decided that the Process Instructions would be created using a word processor and distributed as paper documents as these documents were not changed frequently. It was also decided to continue to use the Change Control Form to request, notify and authorise changes to both the Process and Product Instructions but not to use Change Notes.

The author used the satellite systems approach to design the Product Instructions database, (from now on referred to as the 'Manufacturing Instructions' system as the Process Instructions were not addressed by this work). The database was to have the following characteristics and functions:

- It was to be an indexed database with three types of access; read only, edit and delete;
- There was to be a method to automatically copy old data to a history file as an audit trail;
- The system was to be transportable for use with other MFG/PRO databases.
- The system was to be a self contained system written to operate and be accessed as part of the core.

The key, unique, attribute of the Manufacturing Instructions system is the Item Number. The tooling for the machinery used and the mix are also key attributes but are not unique by item. Unlike the Property Recording system these attributes do not assist with identification of any other necessary data that is stored within the core system.

As a result of there being no interaction between the Manufacturing Instructions system and the core system the decision was made that the two systems would exist and operate as one whole but would not actually be connected.

The new Manufacturing Instructions system consists of three main files, one to store the data, one to store the headings that identify the indexes for the data in the data storage datafile and one to identify the headings for the data stored against the indexes. The files are detailed in the appendices as section 6.4.

Once the headings have all been defined a user can enter, edit and read data in the Manufacturing Instructions database. The system does not cross check Item Numbers against the core system and as a result new items can be entered whilst they are being evaluated and undergoing trials, before they have been entered onto the core system. This is done either by using a new Item Number allocated to the new item or by using the customer reference for the item. If the customer reference is used because there is no Item Number associated to the item, then the index will have to be updated once the Item Number is allocated.

To either enter new data or edit existing data a user first enters the index information about the item in question in the data entry screen. The system checks whether an item, (or range of items), match the data input.

If matches are found the system enters edit mode, selects the appropriate template and displays the data field headings and associated data for editing.

Once the user has finished editing information the system stores the old record in the history file and replaces the data in the main datafile with the new data.

If no matches are found the system prompts for a template number. Based on this template number the system creates appropriate data field headings and invites the user to enter the appropriate data. Once the user has entered the data the system creates a new entry in the main datafile.

The Manufacturing Instructions system is a self contained system with all the required logic operating within the system. The system is not linked to any part of the core system but is stored within the core DMS and operates as part of the core.

Users input data against pre defined index fields to select an item or range of items against which data is to be stored. These indexes contain a link to a template file that defines the data field headings that define the data stored against the index.

As the system is self contained it can be transported to any MFG/PRO core database, irrespective of the options and modules used. Furthermore the system will operate in any environment that uses the same operating system and programming language.

As a direct result of using these systems there was unplanned functionality available within the core system. The concept of a series of integrated systems offering more functionality than simply the sum of its parts is 'emergence', (Checkland 1981). The emergent functionality from the quality systems was the Batch Tracking Report.

3.4.4 Batch Tracking Report

There are a number of circumstances where the component parts of a completed brake pad need to be investigated. For example when it is necessary to track down the customers that received brake pads made from a particular batch of mix material.

Ferodo manufacture mix material from chemicals before pressing the mix along with a metal plate to make a brake pad. The mix is controlled within the mixing section using a four figure mix batch number which does not correlate to the MFG/PRO generated Work Order ID for mix.

The reason for using the four figure number rather than the six figure Work Order ID is that the stamping machinery, which stamps the mix batch number onto a completed brake pad, cannot handle a six digit number, four is the maximum. Rather than replacing the stamp with one that can handle six digits the decision was made to maintain the mix batch numbers in parallel with the Work Order IDs.

A system was needed that would allow the correlation of these two identification numbers. Once the Quality Analysis system was implemented it appeared to be an ideal mechanism for the storage of the mix batch numbers for two reasons:

1. It had spare numerical fields available at the bottom of the screen for storage of miscellaneous data;
2. Data was entered after a Work Order was pressed by reading the quality information off a hand written record. Entering the mix batch number at this time added very little extra workload to the users.

With the mix batch number stored in the Quality Analysis system it was possible, (with some minor modifications made to working practice), to integrate the quality systems even further by writing a report that tracked batches. The report goes in two directions:

1. Using the delivery note number to find the mix batch number;
2. Using the mix batch number to find final destinations and delivery note numbers.

When presented with a delivery note number the system tracks back in MFG/PRO to find the brake pad Work Order and from this interrogates the Quality Analysis system for the mix batch number. Once this mix batch number has been found the system can then find out which other Work Orders used the same batch of mix from the Quality Analysis system and then find the destination of each brake pad manufactured against each Work Order.

The increased functionality does not stop there. By using the Batch Tracking report and the Work Order numbers found thereby it is possible to interrogate the Property Recording system to find out the state of the processes that were used to manufacture the Work Orders. These conditions can then be compared against the data in the Manufacturing Instructions system to verify that the correct procedures were followed.

The flexible design of the systems and the fact that they were implemented as satellite systems helped to make this increased functionality easily available. If, for example, the systems had been developed rigidly as part of the core system, (modifying the core), then there may not have been sufficient spare fields available against the Work Order completions screen as they would have been used for other things.

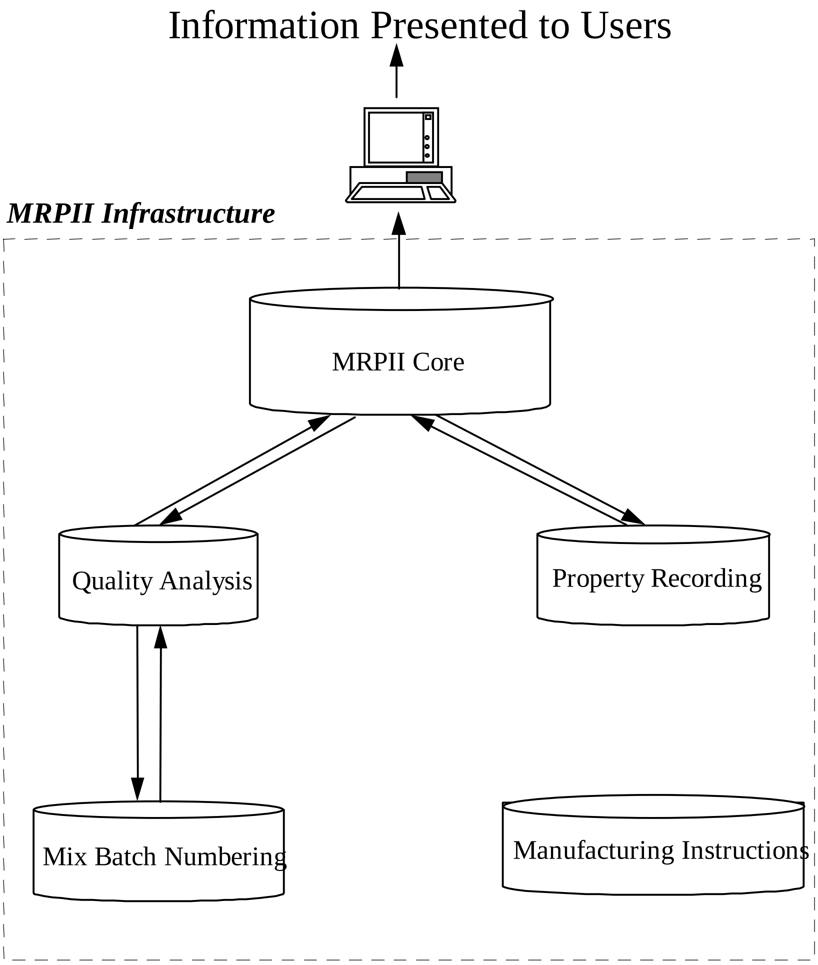
3.5 Development of the Systems Development Methodology

Analysis of the systems integrated with the packaged core system shows that there were three system development techniques used:

1. Incorporation within a satellite system using spare fields, (*Mix Batch Numbering*);
2. Integrating satellite systems with the core, (*Property Recording, Quality Analysis*);
3. Joining unintegrated satellite systems to the core, (*Manufacturing Instructions*).

Figure 26 below shows the layout of these systems and their relationship with the core.

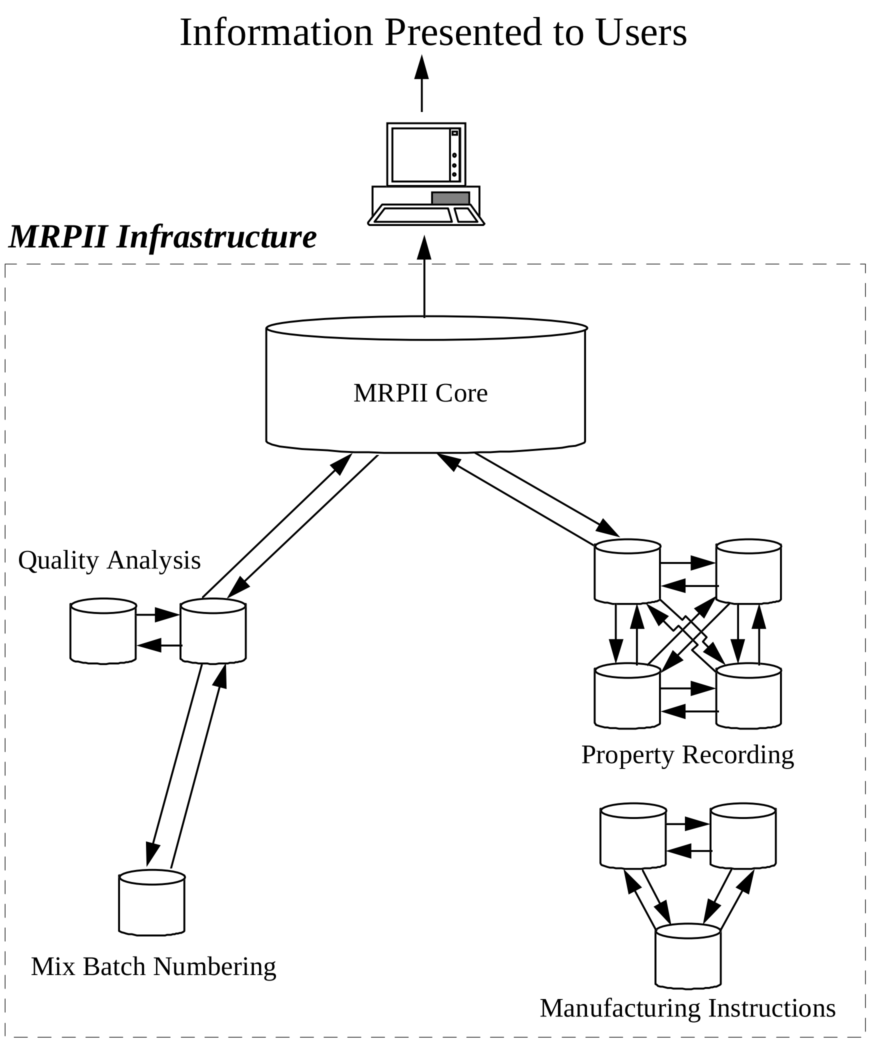
Figure 26: New Systems Database Construction



It can be seen from Figure 26 that at all times users interface the systems via the MRPII core and that the satellite systems follow the hybrid approach to database design.

All of the satellite systems created at Ferodo are distributed databases, apart from the mix batch numbering system. Figure 27 below shows the internal structure of the new systems developed at Ferodo. For the internal structure of the core refer to Figure 13.

Figure 27: New Systems Internal Construction



It can be seen from Figure 27 that the hybrid database structure of Figure 26 is actually a hybrid database comprising distributed databases. Each distributed database forms a cluster that is connected to the core via a simple connection, (if connected at all).

The batch tracking program was described in section 3.4.4. This program collates data from the satellite systems and the core to produce batch specific information for the user. Considering the hybrid system shown in Figure 26 this information was gained using only 3 sets of links between databases. If a distributed database structure was used then it is possible, by inspection, to determine that 6 sets of links are necessary.

In the case of the system shown in Figure 27, however, there are 8 databases that actually need to be linked. If these 8 were to be part of a distributed database then use of the formula derived in section 2.13.2, $n=((m^2-m)/2)$, is necessary to calculate the number of links. In this case it would be $((8^2-8)/2) = 28$ links. If the sub-systems of the core were taken into account then this figure would be extremely large.

In Checkland's soft systems methodology, the perceived view of a system is termed as the Weltanshauung, (see section 2.10). The concept of Weltanshauung applies well to the satellite systems approach to systems design. When designing the mechanism for connecting the satellite systems with the core, the designer needs to concentrate on the systems as a whole, (as shown in Figure 26). The view that the designer takes of the systems is set at an overview level. When designing the systems themselves, however, the designer needs to concentrate on the components of the satellite systems to facilitate connection with the core, (as shown in Figure 27). The view that the designer takes of the systems is necessarily set at a detailed level.

It can be argued that the definition of the core also changes depending upon the perceived view of the situation. When the mix batch numbering system was being developed the Quality Analysis system was used not because it was a satellite system off the core but because it appeared to the system developer as part of a central system. Similarly the batch tracking report was written by a programmer who considered all the new satellite systems to be part of one large system at the time.

The risks being taken when doing this, however, are that satellite systems could be connected with other satellite systems rather than the core, resulting in a confusing hierarchy of systems. Because of this, satellite systems should not be connected to each other. In addition to creating a confusing hierarchy, the resultant systems will become increasingly complex. For example removing one satellite system, for maintenance or obsolescence, will detrimentally affect all other satellite systems that rely on it for connection with the other systems. It is also preferable not to incorporate new systems into satellite systems, (as the mix batch numbering system was), for similar reasons.

For these reasons it is important, when integrating new systems, that the core system is identified as the original purchased package, excluding existing satellites.

When writing reports to extract data from various sources to present information it is advantageous to consider the satellites and incorporated systems to be part of the core.

Analysis of the systems integrated at Ferodo allows definition of three integration techniques. These techniques will be reviewed in section 3.5.1.

3.5.1 The Three Integration Techniques

At the start of section 3.5 the three system development techniques were introduced. This section will explain the characteristics of these techniques, which are listed below:

1. Incorporation within the existing systems using spare fields;
2. Integration with the core;
3. Joining with the core without integration.

3.5.1.1 Incorporation

An incorporated system is one where the data storage requirements are adequately met by the existing systems. There are two pre-requisites for an incorporated system:

1. A similarity in data structure between the requirement and the parent system – isomorphism;
2. Available spare fields within the existing systems.

It is easiest to explain the requirements for similarity in structure through use of an example. If there is a requirement to store data that is created for each batch of product in the factory, then the data can be stored against Work Orders for the batches, as there is one Work Order for each batch and so already a similarity in structure, (a one to one relationship). The data cannot be stored against the Bill of Materials definitions because this is a static datafile that has no relation to dynamic Work Orders or batches of product, (a one to many relationship).

Even if it is possible to identify a similarity in structure it is a fact that there are a finite number of spare fields available within any particular sub-system of a database. For example in all of the satellite systems worked on by the author there are 2 spare fields.

Even if spare fields are available it is necessary to investigate how the fields are used in the company if the new system is to be used elsewhere in the company. It is also possible that the software suppliers will use the fields differently in a future release of the software resulting in conflict when upgrading. As has been seen from experience at Ferodo the spare fields can be from any part of the whole system, including the satellites. Caution must be taken when doing this, however, due to dependence on the satellite system remaining in use, as was explained in section 3.5 above.

The incorporated approach is an integration approach and as such yields the benefits of integration discussed in section 2.12:

1. Reduced keying effort;
2. The ability to verify and cross-reference data, on-line.

These benefits can be best explained using the mix batch numbering example. In this example the data was stored in the Quality Analysis system which was integrated into the Work Order completions screen. By doing this the system already had the Work Order ID number and therefore, the Item Number and Quantity were looked up, removing the need for the user to input this data. The system actually displayed this information to the users to allow them to verify that the Work Order ID number that they are using produces the correct item data.

The incorporated approach, unfortunately, has three drawbacks:

1. Software updates can re-allocate spare fields destroying the incorporated system;
2. The data structure needs to be simple to fit into a finite number of spare fields;
3. The data structure needs to find a match amongst the structures in the main system.

3.5.1.2 Common Characteristics of Satellite Systems

If data storage requirements cannot be met by the incorporated approach, either due to a complex system, insufficient data storage available or lack of a matching structure then the integration or joining techniques can be used. Both the integration and joining techniques involve adding databases to a core system using the hybrid approach to integration and so both are sub-systems, as defined in section 2.14. There are a number of attributes that these systems share. These will be discussed first.

The main characteristics of satellite systems have already been mentioned, they contain data that cannot be handled by the core and they store this data in distributed databases written in the operating language of the core database. These databases are then stored as part of the overall MPC systems. There are two reasons why this may happen:

1. They contain logic, (i.e. data manipulation requirements), not dealt with by the core;
2. They require storage of a large volume of data.

An example to illustrate point 1 is the Property Recording system which required the addition of Statistical Process Control logic to the core. It was evident that there was no mechanism for doing this within the existing core and that a new sub-system was necessary. This sub-system was created as a satellite system.

The Property Recording, Quality Analysis and Manufacturing Instructions systems are examples to help illustrate point 2. All three systems have high data storage requirements which cannot be matched by any combination of spare fields in the core within modules with similar data storage structures.

There are a number of benefits of the satellite systems approach, as follows:

1. System control can be transferred from the computing department to the end users;
2. They use the existing system infrastructure, network and hardware, reducing costs;
3. Using the existing system infrastructure also reduces training requirements;
4. They can be independently updated without the need to take the core off-line;
5. The overall system complexity is reduced due to creating clusters of databases;
6. The systems are self-contained so can be transferred to other core databases;
7. They can be documented separately;
8. Development after implementation of the core allows easier design of systems;
9. Within a large organisation having transportable systems reduces overall variety.

Each of these benefits will be discussed in turn.

1. Satellite systems are designed to be self-contained systems with control files and datafiles controlling operation of the systems. As a result of doing this they are maintained by authorised users. If the programmers defined the operation of the system when creating the software then the programmers would have to maintain the system when requirements change. All the satellite systems used within Ferodo are controlled by the users. The Property Recording and Manufacturing Instructions systems are maintained by staff in the quality department. The Quality Analysis and Mix Batch Numbering, systems are maintained by the OE Business Analyst.

2. The existing core system in Ferodo is accessed across a site-wide network. Extensive training in the operation of the system has already been given to users. By integrating satellite systems with this core there was no need to buy new hardware.

3. As a result of the extensive training given, users are familiar with the operation of the core and its modules. To fully explain this it is worth considering the analogy of Microsoft Windows products. These products are all designed to look and feel the same. Experienced users of one word processing package can easily use another word processing package with minimum training because of similarity in operation. This is true between package types. Users of word processing packages have already learned necessary skills for basic operation of a spreadsheet package.

4. The author's experience of industry shows that business requirements are constantly changing. There will be occasions where an existing system has to be used in a new manner, which may not have been anticipated when the system was designed. A result of this may be that the system needs to be modified. If the system was part of the core this would necessitate making programming changes to the core, possibly requiring users to stop using the core until these changes were made. Satellite systems are self-contained and their removal does not disrupt operation of the core.

5. It has already been discussed at the start of section 3.5 that systems designers can simplify the design process by varying their view of the systems, (their Weltanshauung). The effect of this is to reduce overall system complexity.

6. A result of the satellite systems being self contained is that they can be separated from the core and transferred, or copied, to another database with a similar data structure. Reasons for transferring satellite systems like this may include the merging of databases, the dispersal of common systems within a group of companies or commercial reasons, (such as selling the satellite systems as a product).

Ferodo plan to transport the satellite systems to sister companies within the product group to ensure that all the companies use common solutions to common problems.

7. As a result of satellite systems being self-contained, documentation need not refer to operation of the core. This allows the systems to be independently documented.

8. The satellite systems approach explains that the creation of satellites should follow the implementation of a core. Using this approach assists the designers to identify business requirements by reducing the complexity of the core implementation at an early stage in the overall systems development process.

9. It has already been stated that Ferodo plan to transport the core within the product group to ensure common solutions to common problems. By doing this they can ensure that there is low variety of systems and therefore, experience can be easily shared between the sister companies. Within a large organisation, such as the Friction Products Group, it is not unusual to have a common team of systems specialists supporting the implementation of all the systems. If the variety of systems being supported decreases then their effectiveness at supporting systems will increase as they get more experience of the common systems.

Despite this list of benefits of the satellite systems approach there are a number of problems that can be faced:

1. Satellite systems are time consuming to develop, (*resulting in delays and costs*);
2. Detailed design is necessary, (*also resulting in delays and costs*);
3. Radical changes to the operation of the core can cause connections to be disrupted.

The satellite systems approach, adopted to design the systems at Ferodo, required the design of self-contained systems using control files and data files to allow users to control operation of the system. Designing these systems required careful analysis of the situation being addressed and anticipation of future requirements. It takes much longer to do this than to create a system for current requirements with system control defined in the software. Therefore, increased development costs must be expected and this should be factored into any cost analyses, (which should also reflect the long-term savings to be made from using satellite systems).

If there is a significant change in the operation of the business the satellite systems may lose their connections to the core. For example if a business was to implement Kanban control in part of the factory and completely remove Work Order control then any satellite systems connected to key fields in the Work Order module of the core would be rendered useless, (the system would operate but would have to run independently).

All of the characteristics of satellite system listed above are generic. There are, however, some significant differences between the two available options, integrated with the core and joined with the core without integration, which are explained below.

3.5.1.3 Integrated Satellite Systems

A satellite system that is integrated with the core is one where the satellite system operates as part of the core and the two are logically connected. By being connected in this way the core and the satellite systems can access data in each other database. This system is truly integrated in terms of the definitions given in section 2.12. As a result of this true integration these systems yield the standard benefits of integration, (reduced keying effort and the ability to verify and cross-reference data on-line).

3.5.1.4 Joined Satellite Systems

A satellite system that is joined with the core is a system which operates as part of the core but does not interrogate the core to extract data. These systems are, however, accessed via the core and reports can be run using data from the satellite system from within the core so there is a one-way connection. This type of connection conforms closest to the definition of integration, (rather than linking), given in section 2.12, except that data transfer is one-way. For this reason this type of satellite system has been given the title 'joined' to illustrate that it is integration with one-way data transfer.

It has been shown in this thesis that there are three approaches to integration:

1. Incorporation within a satellite system using spare fields;
2. Integration with the core;
3. Joining with the core without integration.

The author's experience at Ferodo has shown that there are five other system development techniques:

1. Purchasing a new module or core, (*if available from the core system supplier*);
2. Purchasing a separate package and running it independently;
3. Linking a purchased package to the core;
4. Modifying the core;
5. Changing business practice to fit in with the systems available.

It was evident that a tool was required to help system developers decide upon the optimum system development technique to use. The Systems Development Life Cycle Methodology was developed for this purpose. This tool, created by the author from observation, discussion and subsequent refinement, is presented in Section 3.5.2.

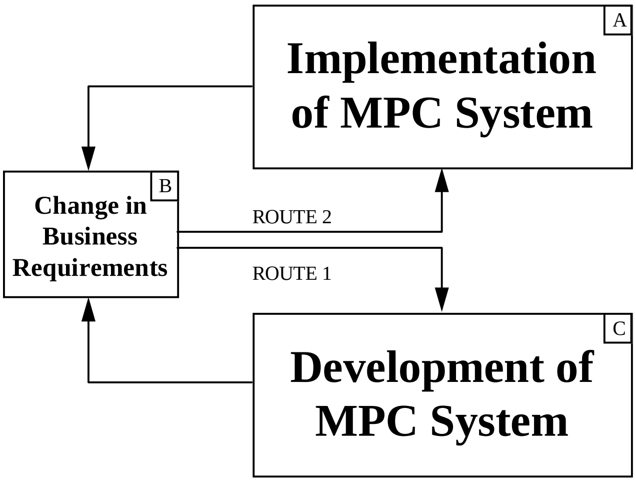
3.5.2 The Systems Development Life Cycle Methodology

Section 2.14 showed that the that the development of sub-systems should be done after the implementation of a purchased, packaged, core manufacturing database. It was also established in section 2.13 that core systems constantly change in response to changing business requirements. There are two ways that these core systems change:

1. Development of the existing systems;
2. Implementation of a new core system.

This is represented in Figure 28 below, starting in Box B.

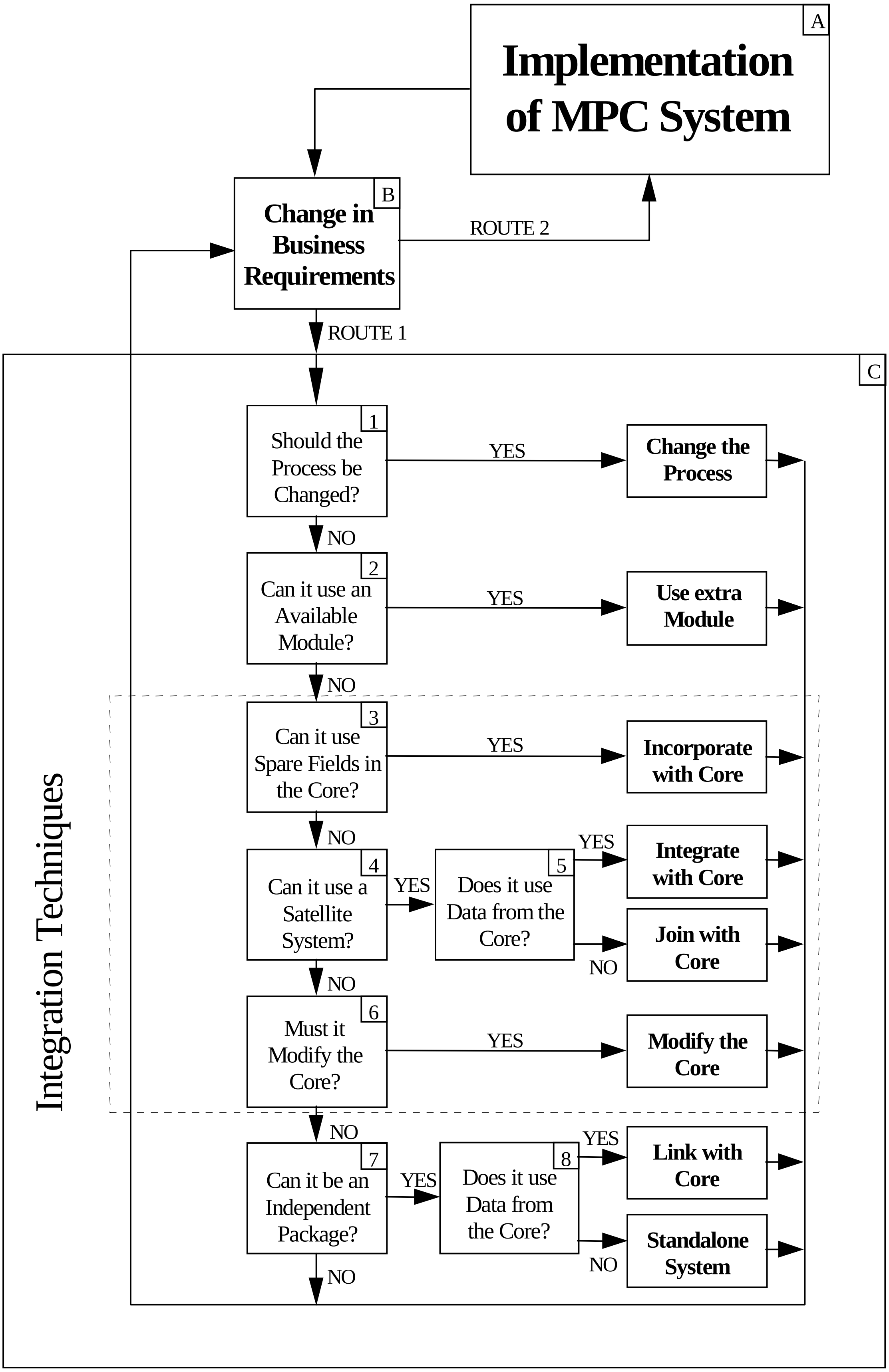
Figure 28: Summarised Systems Development Life Cycle



This diagram shows that the initial stage, Box A, of the life cycle is the implementation of a core manufacturing database as part of a Manufacturing Planning and Control, (MPC), system. When there is a change in business requirements, at Box B, there are two routes available to the systems developers, Route 1, the easier, to develop the existing system in Box C and Route 2, the harder, to purchase and implement a new core in Box A, (implementation approaches were explained in section 2.3 and do not form part of this methodology).

Figure 29 below details the options and priorities of Route 1, through Box C.

Figure 29: The Systems Development Life Cycle



Section 2.10 defined a methodology as 'a structuring of the systems development process'. The Systems Development Life Cycle Methodology is based on the systems development life cycle diagram shown in Figure 29 above. The methodology assists a systems developer to select the appropriate systems development technique in the event of changing business requirements. It is important to emphasise at this point that this methodology will not assist with implementation of a manufacturing database, it is to be used to assist systems development, when there are changing business requirements after an implementation. There are nine different techniques available from two routes as summarised in Figure 30 below:

Figure 30: Techniques Available to Systems Developers

ROUTE 1:	**(Preferred Route - Systems Development)**
Change the Process	Change the process to fit in with existing systems
Use Extra Module	Purchase a new module from the core system suppliers
Incorporate with Core	Incorporate new data using spare fields within the core
Integrate with Core	Integrate a satellite system with the core, (with data lookup)
Join with Core	Join a satellite system with the core, (with no data lookup)
Modify the Core	Modify the core to perform a new, required function
Link with Core	Link an independent package with the core
Standalone System	Create a standalone system with an independent package
ROUTE 2	**(Done rarely, to use new technology, or if Route 1 fails)**
Implementation	Implement a new core using existing techniques

Figure 29 directs the systems designer down a decision making process, starting with a change in business requirements in Box B. Out of Box B the systems designer follows Boxes 1-8 selecting the technique appropriate to the problem being addressed. It may be necessary to apply the Systems Development Life Cycle Methodology a number of times, selecting a number of different solutions to completely satisfy any change requirements. Depending on the resources available it may be necessary to undertake a number of projects at the same time, otherwise a sequential approach may be necessary.

It can be seen that the decision making process is continuous with all techniques leading back into Box B to allow the definition of changed requirements, to then either follow Route 1 again or, if necessary, follow Route 2. This section will continue by explaining each of the techniques listed above and will give examples of how systems designers would have used the methodology to meet requirements at Ferodo.

3.5.2.1 Change in Business Requirements

There are many reasons why businesses require changes to the systems used in their companies. Examples of these reasons include changing customer requirements, new machinery in the factory that requires new planning systems, the desire to reduce lead-times or stock levels and the need for more relevant information from production. Once the need for a change has been identified the systems designer should follow Route 1 out of Box B until they either find a solution or keep on taking the 'NO' option out of Box 7. Route 2 will not normally be followed upon identification of a change in business requirements. A company such as Ferodo will usually periodically upgrade the core system, effectively implementing a new system. All changes that can be dealt with by upgrading to a new system must wait until this time. There have been no issues raised at Ferodo that have not been addressed by one of the two routes available.

In the event that the systems designers know that the core is about to be upgraded and that the new core will deal with the changed business requirements, Route 2 should be taken as the first route out of Box B. For example, Ferodo would like a graphical user interface to MFG/PRO to help compatibility with Microsoft Windows. To meet this requirement Ferodo are waiting for implementation of a future version of the package.

3.5.2.2 Process Changes

Once the decision has been made to follow Route 1, the first technique that should be considered is whether the process being developed should be changed.

There are two contrasting philosophies that businesses may choose to adopt with regards to systems development:

1. Make the software package conform to operation of the business;
2. Make the operation of the business conform to the established software package.

Ideally the business control systems should allow the business to be controlled without modifying the business. Strict application of this rule, however, can lead to unnecessary complexity, particularly in manufacturing organisations where the main focus of development is improved planning and control. Definition of the scope of a required change should identify the necessity of the change. It was seen at Ferodo that minor compromises in business operation yield significant reductions in system complexity.

An example of these minor compromises can be drawn from the original core systems implementation. Before implementation of the core, the part numbering system was based on the customer part numbers. A result of this was that there were many cases where one particular item was sold to a number of different customers with different part numbers. To emulate this in the new core would have necessitated adding an extra level into the Bill of Materials structure to show which items were converted into each customer part number. Ferodo manufacture product in batches. As a result of this it would have become difficult to correctly identify the source and true due dates of batches of items. This is because one batch of items may satisfy a number of customer orders, all with different due dates.

The solution found to this problem was to adopt a generic part numbering system and ask all customers to place demand for these generic numbers, (or convert the numbers from customer to generic using the appropriate MFG/PRO lookup table when orders are received). As a result, there was no requirement for extra levels in the Bill of Materials and the source of demand for each item within the factory is easy to establish. A minor compromise made when implementing the systems vastly reduced system complexity.

Similar compromises can be made when developing existing systems. The author was asked, once the pilot of the Property Recording system was started, to meet a new business requirement. The requirement was that the systems were to be changed to prevent product being despatched before testing was complete. Initial inspection resulted in no possible solution, other than to modify the core systems. This was not acceptable as it would have necessitated a major modification which would have caused problems each time the core was upgraded. It was then decided that the process needed to be changed by implementing manual procedures to prevent Work Orders being completed before testing was complete.

A third example, also with the Property Recording system, is that the initial specification required a graphical display of test results. This could not be achieved with current technology. The choice was to wait for the new version of the core, with a graphical user interface, or remove the requirement. The requirement was removed.

If a simple change can be made to the process to reduce complexity then this should be done by following the YES option out of Box 1 and then changing the business requirement. Otherwise the NO option must be followed out of Box 1 to Box 2.

3.5.2.3 New Module

If the process cannot be changed then the next options that need to be considered are whether to use either an existing module of the core or a new one purchased from the software suppliers. This is represented by Box 2. There are a number of advantages to using modules that are already available for the core:

1. The modules already exist and therefore, time need not be spent designing a system;
2. Expert support will be available from the software suppliers;
3. Purchased modules will integrate with the core and become part of a larger core.

These benefits appear to show that this should be the preferred systems based technique used by systems developers. There are, however, three problems with this technique:

1. Purchased modules can be expensive;
2. The modules available may not meet requirements;
3. There may not be any modules available.

When Ferodo were first deciding upon an approach for the Property Recording system there were no satisfactory existing modules and they decided to look at the MFG/PRO quality module to test its applicability. It was decided that it was too complex and expensive for the limited SPC requirements on-site and another solution was sought.

If a module is available and can be purchased within the budget allowed then this should be done by following the YES option out of Box 2. If, however, there are no satisfactory modules available at the right price then the NO option must be followed out of Box 2 to Box 3.

3.5.2.4 Incorporated Integration

If new or existing modules cannot be used to meet the changing business requirements then the next technique that systems developers should consider is whether to incorporate the requirements into the core. Section 3.5.1 detailed this technique. To summarise, when deciding whether to use spare fields within the core, (or within satellites off the core), the systems developers need to find:

- Part of the core with a similar data storage structure to the data requiring storage – isomorphism;
- Available fields in this part of the core.

Once these fields have been identified the designer also needs to find out:

- Whether future releases of the software will use those fields for another purpose;
- Whether any other part of the company already uses the fields for another purpose.

From the author's experience at Ferodo, an application requiring more than 2 spare fields cannot be addressed by incorporated integration.

Examples of this at Ferodo include the Mix Batch Numbering system already discussed and storage of the ID number of the cell used for manufacture of a particular product against a Work Order. Both of these two requirements were met by using spare fields in the Quality Analysis system as this system stores data sorted by Work Order ID. It has already been shown, however, that using spare fields in a satellite system is not ideal.

If the requirements can be met by incorporating data into the core then the YES option should be followed out of Box 3. Otherwise the NO option must be followed to Box 4.

3.5.2.5 Satellite System Integration

If the business requirements cannot be met by incorporating data within the core then the next technique that should be considered is the creation of satellite systems. Section 3.5.1 explained satellite systems in detail. There are two types of satellite system:

1. Integrated satellites;
2. Joined satellites.

Both types of satellite system are used when there is a requirement for a new system that contains its own data processing requirements or has large data storage requirements, or both. The difference between the two is that integrated satellites look-up data from the core, whereas joined satellites do not look-up data from the core but are still accessed via the core. It has been shown that satellites should not be integrated with other satellites but directly into the core. Examples of integrated and joined satellite systems have already been given:

1. The Quality Analysis system, (section 3.4.1), an integrated satellite system;
2. The Property Recording system, (section 3.4.2), an integrated satellite system;
3. The Manufacturing Instructions system, (section 3.4.3), a joined satellite system.

If requirements can be met by designing satellite systems then the YES option should be followed out of Box 4 into Box 5. If the satellite system requires data to be looked up from the core then the YES option should be followed out of Box 5 and an integrated satellite system designed. Otherwise the NO option should be followed out of Box 5 and a joined satellite system designed. If requirements cannot be met by satellite systems then the NO option must be followed out of Box 4 into Box 6.

3.5.2.6 Core System Modification

If business requirements cannot be met by using a satellite system then the next technique that the systems developers should consider is whether the core systems need to be modified.

As previously discussed, modification of the core is avoided, if possible, by many companies, (including Ferodo). This is because modifications make upgrading of the core more difficult as each modification would need to be made again, (if remembered). There are times, however, where a core modification is absolutely necessary.

Two examples of modifications made to the core Work Orders system at Ferodo are:

1. Checking Work Order quantities against the stocks issued;
2. Quality Analysis screen access.

The first modification was required to help prevent stocks become inaccurate due to incorrect keying. It was noticed that sometimes keying errors were made when entering the quantity of components issued, (removed from stores), against a completed Work Order. The most typical error was entry of an additional zero at the end of a number, increasing the stocks withdrawn by a factor of ten. This caused large errors in stock record accuracy.

In order to stop this happening the core was modified to give a warning message when the quantity entered as withdrawn from stores was more than 25% different from the quantity expected.

The second modification was a method of providing easy access to the Quality Analysis system by using a spare field in the Work Order completion screen. The modification allows a user to press function key F2 to access the Quality Analysis system when they reach a particular spare field in the completions screen.

The reasons for this are so that the user does not have to enter two screens to complete a Work Order and enter Quality Analysis data. Doing this, however, removes a spare field in the Work Order completions screen preventing it from being used for a future Incorporated Integration.

The first modification can be seen to be necessary; there was a requirement for better stock accuracy and this modification helped identify errors whilst not having any other effect on the operation of the core. This modification will have to be made each time the core is updated, until it is adopted by the software supplier.

The second modification, however, cannot be justified. The Quality Analysis system could have been entered as a menu option within the core rather than using a spare field. This modification will have to be made each time the core is updated until such time as the spare field is used by the core, (or by individuals in the business), for another purpose. It is crucial, therefore, that only essential modifications are made to the core as using them as a simple solution initially could cause problems in the future.

If the requirements have to be met by making modifications to the core then the YES option should be followed out of Box 6 and then possibly re-defining business requirements. Otherwise the NO option should be followed out of Box 6 to Box 7.

3.5.2.7 Independent Package

If it has been decided not to modify the core then the last technique that systems developers should consider is to use or to purchase an independent package. There are two ways that this package could be used:

1. Linked to the core, (linking was defined in section 2.12);
2. As a standalone system.

An independent package that uses data contained in the core could be linked, using ASCII data download techniques, to the core. This would be done if the timescale or the complexity of the problem prohibit developing a satellite system due to lack of resources or expertise. Examples of these systems being considered at Ferodo are FCS and Forecasting packages.

An independent package that uses little or no data from within the core could be left standalone from the core, (not integrated or linked to the core). Any data transfer requirements would be so small that they would be done manually. Typical examples of this are Word Processing packages which do not need to share data with the core.

If requirements can be met by an independent package then the YES option should be followed out of Box 7 into Box 8. If the package requires data to be looked up from the core then the YES option should be followed out of Box 8 and the package linked. Otherwise the NO option should be followed out of Box 8 and a standalone system used. If requirements cannot be met by an independent package then the NO option must be followed out of Box 7 back up to Box B and the requirements re-examined.

3.5.3 Systems Development Life Cycle Methodology Decision Support

The Systems Development Life Cycle Methodology defined above requires a user to make decisions based upon their prior knowledge and experience. Use of the methodology is, therefore, limited to practising systems analysts. It was apparent that for it to be applied by inexperienced users an expansion of the decision making process was necessary.

Section 2.12 detailed the systems integration work of Reynolds and Wainwright, (1997). This technique could be used to help determine which integration projects should be undertaken within a business. Once the projects have been chosen then the Systems Development Life Cycle Methodology can be applied to identify the technique to use.

There have been two techniques developed to assist with application of the Systems Development Life Cycle Methodology, designed to be used independently or together:

1. Decision Support Questions, (any negative answer means moving to the next step);
2. Indicative Ratios.

3.5.3.1 Decision Support Questions

The nature of the Systems Development Life Cycle Methodology is of a decision flow, rejection of one option in the decision flow results in immediate transition to the next option in series. A number of simple questions have been developed from experience to assist with each decision to make within the methodology. The questions have been developed so that a negative response to any question signifies proceeding to the next option. Each set of questions relates to a level in the Systems Development Life Cycle Methodology and should be addressed in turn.

Every attempt has been made to keep the questions at a top level and simple. To do this it has been necessary to assume that a degree of business knowledge is possessed by the user. The questions, therefore, can be interpreted as being designed to assist the user to identify the questions they themselves need to address and are presented grouped by the level of the Systems Development Life Cycle Methodology they represent.

Should the Process be Changed?

1 Is it possible to change the process to meet the changed business requirements, (taking into account external accreditation and quality guidelines)?

2 If so, can the new process be used without causing problems elsewhere in the system that cannot be easily addressed by this methodology?

Can it use an Available Module?

1 Is there a module available?

2 Does the module contain sufficient functionality?

3 Is the module affordable?

4 Are the processes within the module simple enough to use, (such as not causing too many transactions, requiring additional staff to be taken on or overburdening existing staff)?

Can it use Spare Fields in the Core?

1 Are there any applicable spare fields in the core, (that store data in the same format, preferably using the same indexes)?

2 If so, are there enough of these spare fields to use for the new system?

Can it use a Satellite System?

1 Can an auxiliary database be written to meet business requirements, (remembering the link into the existing systems)?

2 Is there sufficient programming resource available to do the work?

3 Can the work be done within budget?

Must it Modify the Core?

1 Can modifications be made to the core to meet the business needs?

2 Is there sufficient programming resource available to do the work?

3 Can the work be done within budget, (remembering the continuing cost of each upgrade)?

Can it be an Independent Package?

1 Is there an independent package available?

2 Does the independent package contain sufficient functionality?

3 Is the independent package affordable?

4 In the event of a linked system can the independent package link to the core?

5 Can this be achieved at an affordable cost?

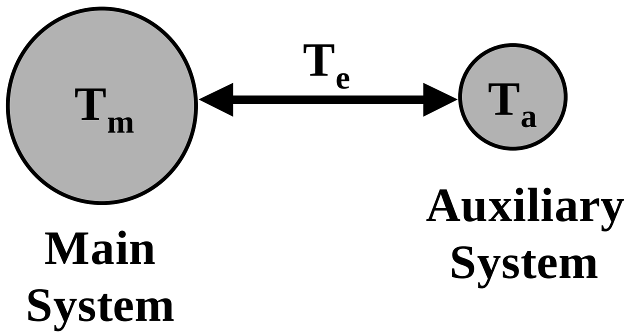
These questions should be used in conjunction with the detailed descriptions of each option in the methodology described in sections 3.5.2.1 to 3.5.2.7 as both the questions and detailed descriptions will assist with the decision making process.

As was stated earlier, the questions above should be used to assist an inexperienced systems analyst decide the questions they should be asking themselves during a systems development project.

3.5.3.2 Indicative Ratios

Both the Systems Development Life Cycle Methodology and the Decision Support Questions adopt a straight questions and answers approach to the decision making process. A series of Indicative Ratios have been developed from experience of systems development to help verify the decisions made when choosing an integration approach from the methodology shown in Figure 29. These Indicative Ratios are based upon the number of transactions that will occur within and between the core and satellite systems. The Indicative Ratios can be illustrated by Figure 31 below.

Figure 31: The Decision Support Indicative Ratios



Definitions
T_m = Transaction rate in main system
T_a = Transaction rate in auxiliary system
T_e = Transaction rate between systems

1. If $T_m/T_a = \infty$ then no need for new system, (as $T_a = 0$, i.e. No Auxiliary System);
2. If $T_m/T_a >> 1$ then Incorporated System;
3. If $T_m/T_a > 1$ then:
    - 3.1. If $T_a/T_e = \infty$ then Separate System / Joined Satellite System, (as $T_e = 0$);
    - 3.2. If $T_a/T_e >> 1$ then Separate System / Joined Satellite System;
    - 3.3. If $T_a/T_e > 1$ then Integrated Satellite System;
    - 3.4. If $T_a/T_e = 1$ then Integrated Satellite System;
    - 3.5. If $T_a/T_e < 1$ then Integrated Satellite System;
    - 3.6. If $T_a/T_e << 1$ then Modified Core.
    - 3.7. If $T_a/T_e = 0$ then Modified Core, (as $T_e = \infty$).
4. If $T_m/T_a = 1$ then New Core;
5. If $T_m/T_a < 1$ then New Core;
6. If $T_m/T_a << 1$ then New Core;
7. If $T_m/T_a = 0$ then Implement Core for first time, (as $T_m = 0$, i.e. No Core System).

Assumption: $T_a \propto T_e$

Use of the Indicative Ratios hinges around two relationships:

1 The relationship between the number of transactions in the main and auxiliary systems;

2 The relationship between the number of transactions in the auxiliary system and the number of transactions flowing between the two systems.

Initially the relationship between the number of transactions in the main system and the number of transactions in the auxiliary system should be established. With the Indicative Ratios this is achieved by dividing the number of transactions in the main system by the number of transactions in the auxiliary system. The outcome of this division needs to be evaluated, compared and applied to levels 1 – 7 of the Indicative Ratios as shown in Figure 31. The author accepts that it is unlikely that any systems development exercise would need to use the Indicative Ratios to show that there is no core or no auxiliary system, options 1 and 7, these options are covered to ensure that the model is complete, (by illustrating the 'boundary conditions').

If the outcome of this exercise shows that the auxiliary system is larger than the core system, then the integration activity will be extremely complex and as such a new core system should be sought, containing the additional functionality required. This is represented by options 4 – 7 of the Indicative Ratios.

If the outcome of the exercise shows that the auxiliary system is much smaller than the core then the most likely 'ideal' systems integration technique would be incorporation using spare fields. This is represented by option 2 of the Indicative Ratios.

If the outcome of the exercise shows that the auxiliary system is smaller than the core then option 3 should be selected. Option 3 is sub-divided into a number of sub-options based upon the second relationship; the relationship between the number of transactions in the auxiliary system and the number of transactions flowing between the two systems. A representation of this relationship is achieved by dividing the number of transactions in the auxiliary system by the number of transactions flowing between the two systems and determining whereabouts within the Indicative Ratios options 3.1 – 3.7 the results fall.

If the outcome of this second ratio shows that the number of transactions between the two systems is extremely small, (or non-existent), then the auxiliary and main systems do not need to be integrated and can either be separate, (independent), systems or integrated using a Joined satellite system. A Joined satellite system approach would be adopted to gain the benefit of a common system infrastructure, such as the Manufacturing Instructions system explained in section 3.4.3 above. If the common infrastructure is not considered to be important then the standalone systems approach would be adopted. This is represented by options 3.1 and 3.2 of the Indicative Ratios.

If the outcome of the second division shows that the number of transactions between the two systems is extremely large, then the core should be modified so that both systems operate within the core, such as the system developed to check the work order receipt quantity against the stocks issued, (as detailed in section 3.5.2.6 above). This is represented by options 3.6 and 3.7 of the Indicative Ratios.

If the outcome of the second division shows that the number of transactions between the two systems is of a similar size to the number of transactions within the auxiliary system then the Integrated satellite system approach should be adopted, such as the Property Recording system presented in section 3.4.2 above.

Unlike the Decision Support Questions the Indicative Ratios technique requires the user to have a level of prior experience to make judgements about exactly how to categorise each eventuality. For this reason it is recommended that it would be better if inexperienced system developers use the Indicative Ratios to help to confirm the systems development technique chosen from use of the Decision Support Questions and the descriptions of each step presented in section 3.5.2, rather than to identify the route to take in the first place. Similarly an experienced systems analyst may choose to use the Indicative Ratios to verify the decisions they made during the decision making process but may also use the technique to assist with decision making in the first place. As a rough guide the Systems Development Life Cycle and the decision support tools should be applied as follows:

1. Select a likely approach from an initial investigation of the methodology;
2. Apply the Indicative Ratios if an integration approach is selected;
3. Apply the Decision Support Questions to verify the decision.

Both of the techniques offered above have been developed by reviewing the outputs from the systems development exercises documented in detail in section 3.4. They have been developed to assist with the decision making process undertaken by system developers as they work through the different stages of the Systems Development Life Cycle Methodology.

3.6 General Applicability of the Methodology

It was shown at the end of section 2.6 that companies using JIT may use an MRPII database if they have large data storage requirements. Section 2.7 described the three main datafiles of the OPT database, datafiles which store similar data to the MRPII database. Due to this similarity in database requirements for businesses using OPT, MRPII and JIT, (in companies with large database structures), the same Systems Development Life Cycle Methodology, as shown in Figure 29, is applicable.

The Systems Development Life Cycle Methodology assists systems developers choose the most effective systems development technique for a manufacturing database. The systems development examples used by this thesis have been applied to an MRPII system, in a discrete batch manufacturing environment using Work Orders to control batches. The satellite systems developed will be transported throughout the Federal Mogul Friction Products Group as all the companies in the group use the same MRPII package as Ferodo and also have Work Order control of batches.

It should be noted that it is not claimed that this approach can be applied to all systems. This is because the methods have been formulated as a result of observation in a restricted set of businesses. One of the key points in this chapter is that the core systems must not be modified unless absolutely necessary. This is generally true for a manufacturing company, where the focus of business is manufacturing and not data processing. In a commercial organisation the opposite may be true as data and system efficiency could be the key concerns.

There is potential for problems with satellite systems as explained in section 3.7 below.

3.7 Problems with the Satellite Systems Approach

Section 3.5.1 described the satellite systems techniques available and some of the problems found with these techniques. The problems discussed were:

1. Satellite systems are time consuming to develop and have associated labour costs;
2. Detailed design is necessary and therefore, also has associated labour costs;
3. Radical changes to the operation of the core can cause connections to be disrupted.

Another problem faced by satellite system designers is documentation. Most of the other techniques in the Systems Development Life Cycle Methodology involve using existing systems in a new way or purchasing new systems. These systems all come with detailed documentation of software operation, documentation is also provided about the structure of the databases. Systems developers have to provide documentation to support the operating procedures for the system but not for the underlying software.

When using the satellite systems techniques or modifying the core a new system is being designed or an existing system is being modified. For these systems the systems developers have to provide documentation to support the operating procedures for the system and also for the underlying software. The result of producing comprehensive documentation is that the systems take even longer to develop.

The next section, section 3.8, will look at the documentation of MRPII systems. This will be followed by a review of a documentation exercise undertaken by the author at Ferodo. The chapter will then further expand the systems development methodology to help authors ensure that documentation is comprehensive. The methodology defines four levels of documentation to assist with documenting satellite systems.

3.8 Structured MPC System Documentation

In section 2.3 the open structure of the database of commercially available MRPII systems was discussed. This structure results in an MPC system that can be easily tailored to work in a variety of organisations as shown in section 1.4.

An alternative approach is to write a system especially for the business in question. The drawback of doing this is that a system is developed that has not been tested elsewhere, cannot necessarily be used when business requirements change and may not have the detailed expert support typically enjoyed by a packaged system. As a result of the problems of writing MRPII systems in-house, most companies buy packages and tailor them to their requirements by selecting modules, (as discussed in section 2.3).

Unfortunately there is a drawback of using and tailoring an MRPII system. The problem is that the operation of these systems is not defined by the software, (Kamenetzky 1985) but is defined by the database configuration and operational practice. Documentation created when the software was written is only useful for software support and so companies need to create their own operational documentation.

Documentation requirements also come about from the satellite systems approach where systems are designed to be used with a variety of databases and with users controlling system configuration, as was shown in section 3.7. This section will propose the use of electronic documentation to assist with the creation of comprehensive documentation within a manufacturing environment. Developed from this, the Systems Development Life Cycle Methodology will be expanded to include the development and installation of documentation.

3.9 The Ferodo MPC System Documentation

The author documented the operation of the OE MFG/PRO database as part of a familiarisation exercise at the outset of the project.

It took 3 months to complete this documentation exercise. Unfortunately, to make the report readable, a number of restrictions had to be imposed on the content:

- It was assumed that the reader already had experience of MFG/PRO and did not need a detailed explanation of the actual physical actions required to accomplish certain tasks;
- It was assumed the reader only needed documentation of the screens that updated or altered the OE database, as opposed to the multitude of report screens;
- Of these screens it was assumed that the reader only needed information about the menu choices that were to be changed, those left unaltered were not to be documented;
- It is assumed that the reader already understood why the procedure was carried out;
- It was assumed that the reader would be satisfied with a summary style report.

Even with all these restrictions imposed, it was only possible to condense the document to 140 sides of single line spaced typed script with an 11 page index. The documentation was then photocopied 18 times and distributed with change control in place to prevent unauthorised copies being made, (a red signature was required on a registered copy).

It was evident when writing this documentation that there were three potential readers of this documentation:

1. Readers interested in learning about the flow of items through a product line;
2. Readers interested in learning the tasks involved in doing a person's job;
3. Readers interested in learning the physical actions to take to do a particular task.

It can be seen from the list of assumptions that it had already been decided not to cover item 3 in the documentation created at Ferodo. In order to allow both the remaining uses, the document would have required two sections, one indexed by product line, the other by operator. This would effectively double the size of the document which would have become too large, (even with the restrictions) and so only one section was written, indexed by operator.

An investigation was made into the size of document required to fully meet requirements. It was estimated that to fully document the use of the OE database, without any of the restrictions above and with documentation of physical actions, the document would be ten times its original size, 1,320 pages long with a 110 page index.

To fully document the use of the databases within each division in Ferodo these figures would be multiplied by a factor of 5 to 6,600 pages and 550 pages of index. With an 18 person distribution list the number of pages that would need to be controlled would have risen to 128,700.

The size of these documents would obviously have caused a lot of paper handling and distribution problems. On a more fundamental level, however, the 'simple' 130 page document produced was hardly used within the OE division. The reasons quoted were that it was too difficult to find the right information and it was physically too big.

3.10 The Ferodo Intranet

The purpose of any documentation system is to allow a reader to easily find and read current documentation. It has been shown that in order to comply with ISO procedures these documents should be created and maintained in a structured manner and must contain a change control and notification procedure. It has also been shown that the Ferodo documentation system was paper based and difficult to manage. Examples given were the original Manufacturing Instructions system, (section 3.4.3) and the MRPII documentation exercise performed by the author, (section 3.9).

As a result of the problems experienced with these systems, the author developed a documentation system that had the following attributes:

- Document creation either directly in the system or by copying from another package;
- Structured document storage;
- Easy document distribution and retrieval;
- Full control over document changes;
- Notification of document changes to pre-defined distribution lists;
- Creation of history records that explain the current revision of a document.

The approach chosen was to create a Lotus Domino Intranet. Lotus Domino is a developed version of Lotus Notes, a groupware application which incorporates structured databases for documents with security and user generated change control procedures. The increased functionality that Lotus Domino offers over Lotus Notes is that once documents have been created they are converted into documents that can be read by a standard web browser directly from within the Lotus Domino system.

The system, designed by the author at Ferodo, was implemented, (under the author's direction), within Lotus Domino by an experienced programmer to cope with a variety of documentation types and applications. The system was designed to allow the user to control the detailed configuration of the final system through selected options.

Each Lotus Notes database can contain a variety of document templates. A template can be likened to a blank page of infinite length with a hidden document control header. The blank page can hold textural or graphical information and can be filled in by cutting and pasting from existing documentation. The Ferodo Intranet documents are all created using the same pre-defined template to ensure common and standard documentation. The hidden header contains the following information:

- Title;
- Index headings, (*stored in the system the first time an index is used for future recall*);
- Creator name, (*automatically captured when the document is first created*);
- Date of creation;
- Document owners, (*entered from a pre-defined list of names*);
- Revision number, (*automatically incrementing, starting at 0*);
- Reviser name, (*automatically captured when the document is revised*);
- Date of revision;
- Revision reason, (*a required field that must be entered when editing a document*);
- Keywords, (*entered from a pre-defined list of keywords*);
- Personnel to notify of changes, (*entered from a pre-defined list of groups of staff*);
- Document type, (*document or form*).

With information in the header a document can be controlled completely as below.

The index headings are used by the system to create a hierarchical document display on-screen to facilitate navigation between documents.

The keywords allow for the entry of separate indexes and for fast searches for documentation using the keyword search facility. There are a number of views that can be used to look at a database using the Intranet template. One way is to index by keyword. The system will display all the keywords in use in all the documents in the database and arrange the documents with these keywords.

For example a document about processing sales orders that is used by the sales and planning departments would have two keywords, ‘sales’ and ‘planning’. A keyword index would show the document in both the sales and the planning sections - two copies of the document on-screen but only one actually stored. In this way the departmental copies of the Manufacturing Instructions system, (section 3.4.3), are emulated.

The ‘personnel to notify of changes’ list is a pre-defined list of groups of EMail addresses. When a document is created the creator should look at the target audience of the document and enter the EMail addresses accordingly. In business, people change jobs frequently. If an individual changed job, then every document with their EMail address on would have to be changed to reflect the new EMail address, (unless EMail addresses were job rather than person specific). To combat this in Ferodo, where addresses are person specific, group names are used. The sales orders document above would have two change notify groups - ‘sales’ and ‘planning’. These groups themselves are then split down, in a central database, into the individuals in the group, (a group may be a group of one if necessary).

When an individual changes job then only the database holding the group profiles need to be altered to reflect the changes, rather than every document in the Intranet.

Finally the creator needs to tell the system whether the document is a document or a form, this information is needed if a document is printed, for document control reasons.

When a document is revised, the reviser is required to enter a reason for the change to create an audit trail. Once the change is complete and authorised the system will send an EMail to everyone in the list of personnel to notify of changes. The EMail message will contain the document name, the revisers name and the revision reason.

There are two types of user of the Intranet:

1. Reader;
2. Author / Editor.

<u>Reader</u>

To access the Intranet the reader opens a web browser on their PC, which will start at their site ‘Home Page’, a starting point for other pages. From here they will be able to follow hypertext links by clicking on the area of interest. For example if they are trying to find out about an MRPII operating instruction they will click on the MRPII button or highlighted text. Using this method of ‘browsing’ it is easy to find the appropriate information by using what is in effect a hierarchical index structure. If they are not familiar with the indexing structure then it is possible to search for documents that contain words of interest, despite the location of the word within the document, by using the search button. This is called full text searching.

When a Lotus Notes document database is accessed by this method, a reader can then view the list of documents in one of three ways; by Title, Index or Keyword and select interesting documents as appropriate. When a document is being read, the reader can be assured that they are reading the current version of the document. If the reader is deemed by the document authors to be part of the approved distribution list then they will also receive an EMail message alerting them to any change as it is made, as discussed above.

There will be instances where a change to a document is required by a reader. In these circumstances the document can be printed out and amended before being sent back to the document owners, it can be copied electronically to a word processor, amended and sent electronically to the document authors to modify the original, or it can be pasted and edited in an EMail editor if it is textural in nature. This effectively closes the loop, providing feedback from the system users.

There will be instances where a document needs to be printed. When a document is printed it will appear with a footer stating that the document is only valid when read from the Intranet. It is important, therefore, that the personnel to notify of changes list contains the names of people likely to require printed copies. If the document was identified as a form then the footer is not printed. This is because forms need to be valid as they may have to be printed and used, (for example an accident report form).

Editor /Author

The editor / author can read the Intranet and browse in the same way as above but can also edit documents. Each editor / author needs access to a copy of Lotus Domino as no editing can be done over a web browser.

The difference between an editor and an author is that within each database an author can create documents but only amend those that they originally created. An editor can create, amend and approve any document, regardless of who created it. The procedures followed within the Ferodo Intranet are as below.

Creating a New Document

When creating a new document the editor / author opens the appropriate documentation database and clicks on the 'New Document' button. This opens a clean page, (in a pink colour), automatically fills in the name of the author and the creation date and sets the revision number to 0. The author / editor then needs to select keywords and groups of people to notify of changes, from the centrally managed list of approved keywords and group names, (if the appropriate words or group names do not exist then the system manager needs to be notified to add them to the list). Once the keywords are defined a title should be entered and an optional index structure created if the database structure requires it. The index headings are selectable from a self-maintaining list.

Finally the editor / author can enter the document by either typing directly into the space provided or they can cut and paste from existing PC based documentation. When finished the save button immediately publishes the document for general viewing.

Amending a Document

When amending an existing document the editor / author opens the appropriate documentation database and double clicks on the appropriate document.

This opens the document and displays two buttons on the top, (if the author has editing rights). Clicking on the 'Change' button converts the document to editing mode, (in a blue colour). The revision number is incremented and the name of the changer is copied into the header. All necessary changes can now be made by either typing directly into the space provided or they can be cut and paste from existing PC based documentation or an EMail message. When finished the change reason field is filled in.

Pressing the 'Save' button puts the document into a hidden file called the Pending Documents file. The editor / author, or their supervisor, can now access the Pending Documents file and authorise the document by pressing the 'Accept Change' button. This overwrites the existing document, sends EMail messages of the change reason to everyone on the change notification list and copies the change reason, date and changer name into another database as change history. Documents can be created and edited by cutting and pasting from existing electronic documents, in addition to manual keying.

To summarise the operation of the Intranet documents:

- The change history is automatically maintained;
- The system will automatically send notification of changes to selected personnel;
- It is possible to view documents by index, keyword or title;
- There is a centrally maintained list of keywords;
- The databases can be searched from anywhere in the Intranet for important words;
- Hypertext links can be created to link documents together;
- Amended documents need approving before release giving extra control if necessary;
- Database structures are user defined and configurable by index;
- Security is defined by Lotus Domino so the Intranet can be as secure as required.

The previous discussion of Intranets highlighted a concern with documents moving location and hypertext links between them being lost. The Ferodo Intranet controls this procedurally by using separate self contained databases for each document controller. For example the MRPII procedures and the Quality Manuals are stored in physically separate databases, linked via the Ferodo Home Page. Within the Quality Manual documentation system there is another split of 10 databases each documenting separate product lines. Each of the 10 product lines has a member of the quality department appointed as document controller.

The author has advised Ferodo to adopt the policy of not creating hypertext links between databases, only within databases. In this way there is a greater chance of a document reviser knowing which hypertext links are affected when a document is moved or deleted. Even if they do not know, the document controller for the database will be able to check the affected links when they approve a document. If a link is required to a document stored in another database the document creator uses keywords to identify the document and the reader simply has to undertake a keyword search to find the document of interest.

The first documents put onto the Intranet were the original MFG/PRO system operating procedures created by the author after the early documentation work. Following on from this the quality documentation was entered, starting with a pilot implementation using two product lines first. The MFG/PRO and quality documents were chosen because they already existed and had change control in place. Starting creation of an Intranet with existing documentation is in keeping with the views of Desborough, (1996).

3.10.1 The Ferodo Intranet for MRPII Documentation

The paper based MRPII documentation prepared by the author, as illustrated in section 3.9, was created through a process of interviewing individuals and so the natural structure of the documentation was sorted by individual. The documentation was converted into an Intranet by splitting the documentation down into logical sections. The Intranet was structured by individual, having been created from the original paper based documentation. Each individual's section was split off from the main documentation and stored as a fully separate document on the Intranet.

Lotus Notes allowed logical views of the data to be created so that readers could see the data presented in a number of formats. All the security and document retrieval benefits of the Intranet, as detailed in section 3.10 above, were obtained.

Within each document hypertext links were created enabling a reader to browse through similar functions or follow the flow of information through a product line.

It was observed that the documentation created in this way met the requirements for a documentation system except that there was still no documentation of the physical actions to take to accomplish all the tasks. There was also no documentation that explained system operation at the top level, (explaining the reasoning behind the procedures followed). The documentation was concentrated at an operational level as a result of only interviewing users and documenting the tasks these users performed. It was evident that a methodology for creation of documentation was required to ensure thorough documentation. Section 3.11 will explain the methodology that was developed.

3.11 Further Development of the Systems Development Methodology

The electronic documentation systems reviewed and applied in this thesis provide a mechanism to effectively address the document control and distribution problem. They do this by providing document creators with an easy to use method for control of any number of documents and by allowing effortless location of the controlled documents. This is done by using either a Document Management System, (a push approach), an Intranet, (a pull approach), or a hybrid that attempts to combine the best of both, such as Lotus Domino.

There are two ways that documents could be created:

1. Centralised documentation, (*one author, many readers*);
2. Distributed documentation, (*many authors, many readers*).

Centralised documents are created and maintained by a dedicated document creator and distributed to all the necessary personnel. In the case of distributed documents, staff create and maintain their own documents. Whichever type of documentation approach is used the documents need to be effectively managed and distributed. The hierarchical indexing of electronic documentation systems assists with this documentation and the use of a standard document format ensures a common look to corporate documentation. Understandably these systems have proven very successful within businesses.

The work done applying an electronic documentation system within Ferodo met all the technical requirements for document storage and distribution. The technology and techniques used within Ferodo to achieve this result could be used by any organisation.

This technology does not, however, assist the document creators with initially creating the necessary documentation. The reason for this is that the technology has been designed to work in any organisation and therefore, is not specific to manufacturing.

It has already been demonstrated in Chapter 2 that there is a similarity between businesses and the MPC systems that they use. It is possible to take advantage of this similarity to create a generic template to assist with the creation of MPC system documentation for manufacturing companies. This is similar to the aims of the ISO organisation which established the ISO procedures and associated documentation to help document creators provide comprehensive quality documentation. There are a number of advantages to using a common template to help create documentation within manufacturing businesses:

1. Use of a common template allows system documenters to develop the template to create a specific map of the system;
2. By developing this map the system documenters can easily create all necessary documentation by 'filling in the blanks';
3. By creating a map from a general template the system documenters can easily identify any errors in the system configuration;
4. If a common documentation structure is used by different companies then it will be easy for individuals, such as consultants, to understand the systems in other companies. This is analogous to the MRPII structure which is common across packaged systems so that, once an understanding is gained of one system, individuals can easily understand another system.

Section 3.11.1 presents the MPC systems template developed from this research.

3.11.1 The MPC System Documentation Template

Analysis of the documentation produced by the author at Ferodo and the functionality of the MRPII system used by Ferodo and the other case study companies, has revealed that there are essentially seven MPC functions, performed by manufacturing companies, that require documenting:

1. Business accounting;
2. Systems management, (*systems administration and database management*);
3. Purchasing, (*supply management and demand management*);
4. Processing customer demand, (*demand management and product despatch*);
5. Manufacturing, (*planning and operations*);
6. Quality management;
7. Stock control.

Experience has shown that many companies are adopting the principles of the focused factory as advocated by Skinner, (1974). These factories are broken down into smaller units with localised management that still operate as part of a large entity. Skinner defined five major characteristics of a business that could result in adoption of a focused factory:

1. Different process technologies;
2. Different market demands;
3. Different product volumes;
4. Different quality levels;
5. Different manufacturing tools.

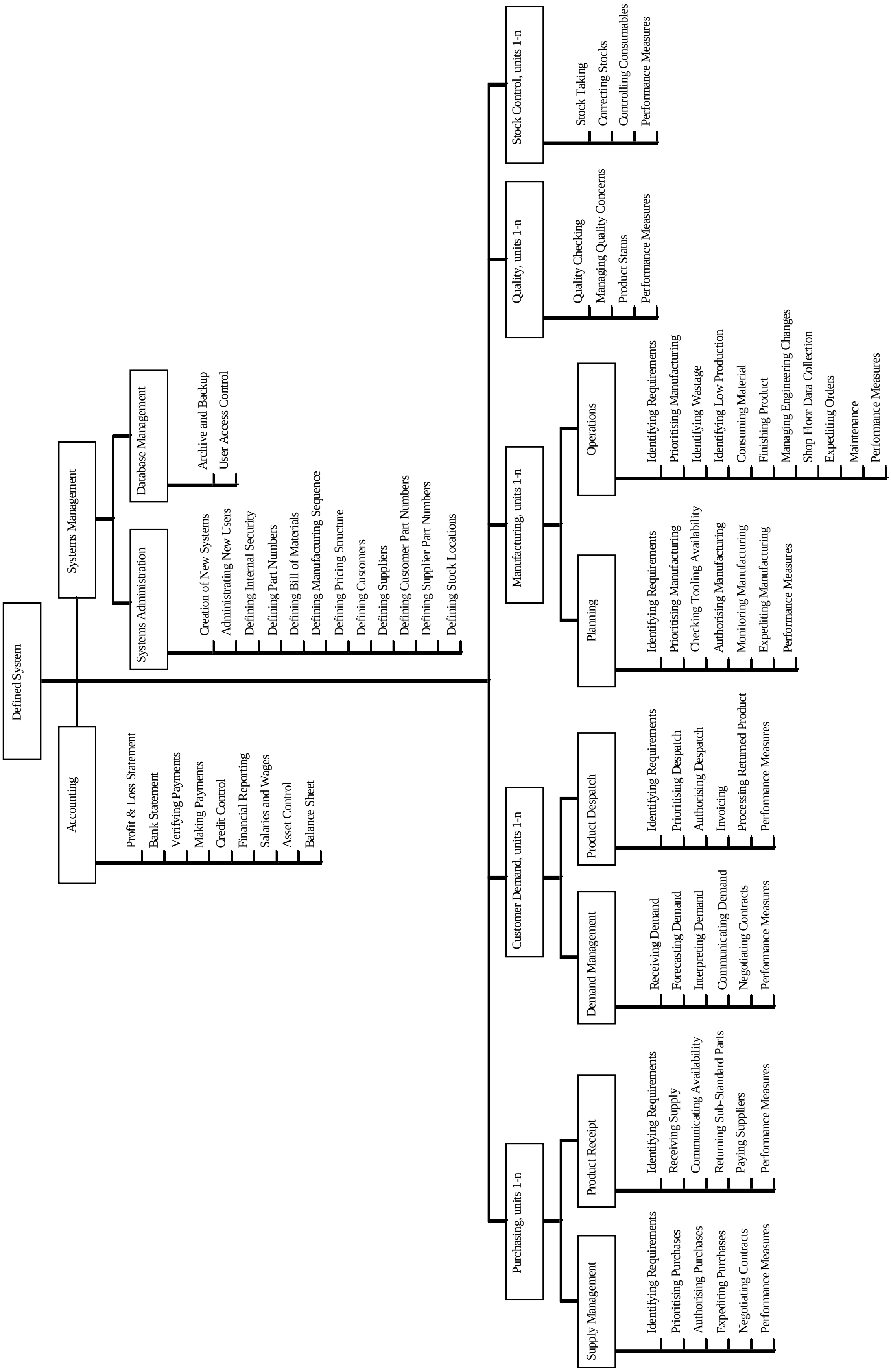
In some circumstances analysis of the business may lead to the formation of a divisionalised company with all the divisions operating as totally separate autonomous units. This happened within Ferodo causing the split between the OE, Aftermarket and Railways divisions. It was decided that the divisions differed too much to operate as one entity, (moreover, splitting the business in this way allowed greater clarity).

In other circumstances analysis of the business may lead to formulation of product lines within the business, or within a division. This also happened at Ferodo, the OE division was further divided into Car Disc Brake Pad, Linings, Commercial Vehicles and Mixing, (Mixing feeds mix into the OE product lines and is a supplier for all divisions).

These product lines share common senior management, systems management and accounting functions but planning and control varies across product lines. There are other ways that a company could be split, however, grouping common functions together. For example at Ferodo the stores and purchasing functions are common across the product lines but are split into two sub-functions to cater for purchasing chemicals and purchasing ancillaries and plates.

It was evident that any MPC system documentation template would have to cater for businesses that are split up into product lines and sections. The template presented in Figure 32 was developed for this purpose. The template was developed from the seven functions of a business defined at the start of this section. These functions were hierarchically broken down into their main attributes by examination of the case study companies, a detailed review of the Ferodo systems, theoretical understanding of MRPII databases and from testing business examples against the template.

Figure 32: Top Level MPC System Documentation Template



The template places the two common functions of an MPC system, accounting and systems management, at the same upper level of the hierarchy, as experience has shown that they usually operate across all the product lines of a business. This level is one level higher than the other five functions of an MPC system, which can themselves be split into a number, (1-n), of different sections. A hierarchical format for this template has been chosen rather than a list because it allows the template to be broken down into a number of understandable groups.

This template has not been designed to prescriptively dictate all the documents that are necessary to comprehensively document an MPC system. The template has been designed to allow documentation creators to select and modify the attributes that best match the MPC system used in their business. It is envisaged that there are two reasons why this template would require modification to suit a particular MPC system in a particular business:

1. If the business does not operate in a discrete batch environment;
2. If the business does not use all of the attributes of the template.

The format of the template should not change significantly when modified to match an MPC system and should, therefore, be used as a model for the structure of the resultant documentation.

Section 3.11.2 will show how this template can be used in combination with four levels of documentation to provide comprehensive documentation. Section 3.11.2 will also give a detailed example of the use of the template and the four levels of documentation to document chemical procurement within the Ferodo OE division.

3.11.2 The Four Levels of Documentation

In section 2.16 two different authors' views on multi-level documentation were reviewed. McLeod *et al*, (1995), show that there are four levels in a business that need documentation:

1. Business operation, (*organisation*);
2. Department operation, (*function*);
3. Employee operation, (*application*);
4. System operating procedures, (*module*).

Hussain and Hussain, (1995), also promoted documentation in four basic types:

1. Aims and objectives, (*systems manual*);
2. Program detail, (*programmers manual*);
3. Program operation, (*operators manual*);
4. Operating procedures, (*users manual*).

These authors define the documentation required to understand the operation of a business, (McLeod *et al* 1995) and its systems, (Hussain and Hussain 1995). It is a combination of the two that is required to adequately document an MPC system. MPC systems assist with control of the business, therefore requiring McLeod's documentation levels. They can also be complex systems, therefore requiring Hussain and Hussain's documentation levels.

In order to adequately document an MPC system, therefore, it is evident that a hybrid of these definitions is required.

The above definitions can be re-arranged into four logical groups of documentation:

1. Business operation, system aims and objectives;
2. Department operation;
3. Employee operation, program operation, operating procedures;
4. System operating procedures, program detail.

Groups 1 and 2 provide elaborate documentation and groups 3 and 4 provide minimalist documentation, (see section 2.16.2). These four groups can be described as below:

1. Documents that describe the **Purpose** of the system;
2. Documents that describe the **Interaction** within and between departments;
3. Documents that describe the **Procedures** followed within departments;
4. Documents that describe the **Software** used to control the business.

These four levels can be conveniently summarised by the acronym 'PIPS'. The software level of documentation of PIPS is normally proved by the software suppliers, with the software package. When creating new satellite systems, as in Chapter 3, however, this software documentation does not exist and needs to be created by the systems implementers or documenters.

Analysis of the work undertaken to provide documentation in the Ferodo OE division shows that it only delivered the basics of the interaction and procedures levels of the above list. This was due to the assumptions and simplifications that were made. There are two approaches that could be used to create the full set of documentation:

1. Work down the PIPS structure;
2. Initially define all parts of the MPC system and then apply PIPS to each part.

Working down the PIPS Structure

Care is necessary when documenting an MPC system by working down the PIPS structure, to ensure that the documentation is comprehensive. The documentation would be created as below:

- The purpose of the whole system and its individual parts would initially need to be documented;
- Once the purpose level has been documented then the interaction between each individual part of the system would need to be documented;
- Next, the detailed procedures for operation of the system in each part of the system would need to be documented;
- Finally the software used to control and run the programs used in each part of the system needs to be documented.

This approach appears very straightforward. There is, however, a real risk of incomplete documentation as it will be difficult, in practice, to identify each part of the system when creating documentation.

For this reason the second approach, as below, would appear to be preferable.

Applying PIPS to predefined parts of a system

To apply PIPS to predefined parts of a system requires the system document creators to initially create a hierarchical map of the system. This map would start at the system in question and break this system down into top level components. These components would then be split down themselves. This stage should already exist, as the modified template developed as in section 3.11.1 would yield this developed system map.

The example below is provided to explain how the modified template is created. The example is based on Chemicals Procurement, (Supply Management), within Ferodo OE.

1 The MPC system in question is the Ferodo OE Division MRPII package.

2 The Ferodo OE Division contains the following multiples of functions:

2.1 2 × Purchasing, (*Chemicals, Ancillaries and Plates*);

2.2 3 × Customer demand, (*Commercial Vehicles, Car Pads, Lined Shoes*);

2.3 3 × Manufacturing, (*Commercial Vehicles, Car Pads, Lined Shoes*);

2.4 1 × Quality;

2.5 2 × Stock control, (*Chemical Stores, Ancillary and Plates Stores*);

2.6 1 × Accounting;

2.7 1 × System management.

3 The Chemicals Purchasing Function comprises the following sections:

3.1 Supply management;

3.2 Product receipt.

4 The Chemicals supply management section, (the Chemicals Procurement section), contains the following sub-functions which use the MRPII package:

4.1 Identifying requirements;

4.2 Prioritising purchases;

4.3 Authorising purchases;

4.4 Expediting purchases;

4.5 Negotiating contracts.

This map has been shown above for one part of the system. Once created for the whole system then PIPS documentation can be created for each part of the system, as below.

Initially the purpose of the MPC system in question should be defined. At this point there is only one entity and so there can only be documentation of the 'purpose' category. For example, the Ferodo OE MPC system manages the manufacturing, planning and control of all friction product items that are made on-site for assembly onto new vehicles.

The MPC system can be split into a number of functions, (departments). For each of these functions the PIPS levels can now be applied. Each function has a purpose which needs to be documented. Each function interacts with other functions and this interaction also needs to be documented. At this level in the map the functions are logically split and have no collective procedures or software requiring documentation. For example, the Ferodo OE MPC system contains the chemicals purchasing function. The purpose of the chemicals purchasing function is to place orders for purchased chemicals. The chemicals purchasing function interacts with the manufacturing functions to monitor demand and the chemical stores to monitor stock levels.

All the functions within the MPC system can then be split into a number of sub-functions. These sub-functions have a purpose, which needs to be documented. The sub-functions interact with other functions and with other sub-functions and this interaction needs to be documented. These sub-functions may be at the bottom of the hierarchy and may also need procedure and software documentation. For example, the Ferodo chemicals purchasing function contains the chemicals procurement section. The purpose of chemicals procurement is to order chemicals from suppliers. The sub-function interacts with the same functions as the purchasing department and with other parts of the purchasing function. Chemicals procurement is at the bottom of the hierarchy, has detailed procedures and uses software, both of which need documenting.

In order for this second approach to work the map of the systems must first be created. This, however, can be made simpler by modifying the template presented in Figure 32.

The template presents the typical operations and functions performed by a manufacturing company making product in batches and controlling the manufacture of those batches. This template needs to be redrawn each time it is applied to a company to reflect actual operation.

Many of the functions presented in this template are operational functions and are the lowest level in the hierarchy. Some functions, however, may need to be further split to be better applied to a company. An example of this could be the control of consumables which is a sub-function of stock control. There could be some consumables used in the manufacturing process and there are stationary consumables. These different consumables may be controlled separately and therefore, need to be documented separately.

In the majority of cases, such as Ferodo, it is possible to derive this structure by inspection of the template and observation of actual working practice. In a very complex environment, however, a tool such as IDEF0, (reviewed in section 2.11), may need to be used. Once the top level of the system is mapped using IDEF0 then the map could be functionally decomposed to produce an detailed map of the systems.

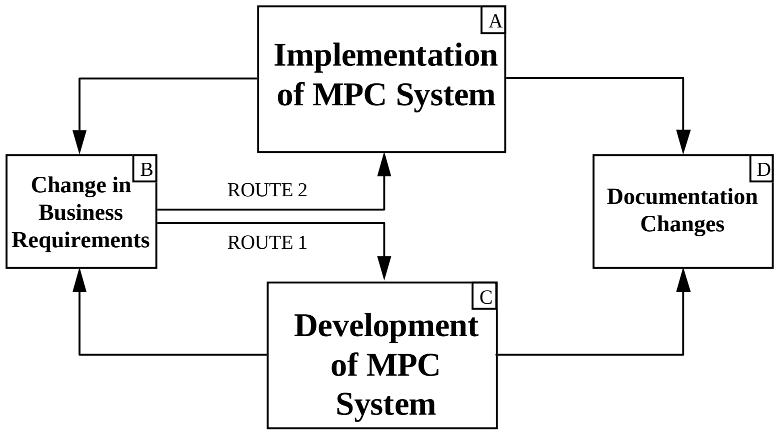
It became evident that a tool is necessary to help guide document creators through the process of document creation. The Systems Development Life Cycle Methodology was further developed for this purpose. This is presented in section 3.11.3.

3.11.3 The New Systems Development Life Cycle Methodology

Section 3.5.2 presented the Systems Development Life Cycle Methodology. This methodology is based upon the Summarised Systems Development Life Cycle which was presented in Figure 28.

Analysis of existing documentation and experience of the documentation exercise undertaken at Ferodo has led to definition of the New Summarised Systems Development Life Cycle shown in Figure 33, which is a development of Figure 28. This diagram shows that documentation is usually produced as a result of MPC systems development, (rather than as part of the process).

Figure 33: The New Summarised Systems Development Life Cycle



This New Summarised Systems Development Life Cycle shows the two ways that systems can change, (implementation and development, Boxes A and C). These changes feed back to Box B to allow future changes in business requirements. In addition to this, however, changes in the operation of the systems drive documentation changes, as shown by Box D.

An effective documentation exercise can also highlight errors in the manufacturing systems with observed errors fed back from Box D to Box B as in Figure 34.

Figure 34: New Summarised Systems Development Life Cycle With Feedback

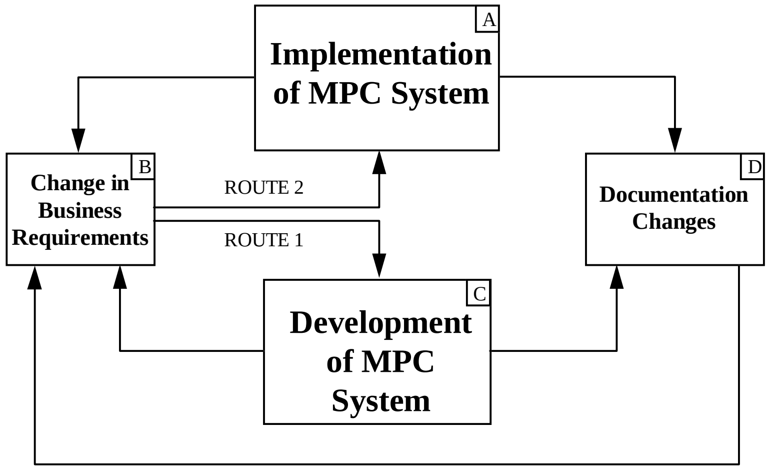
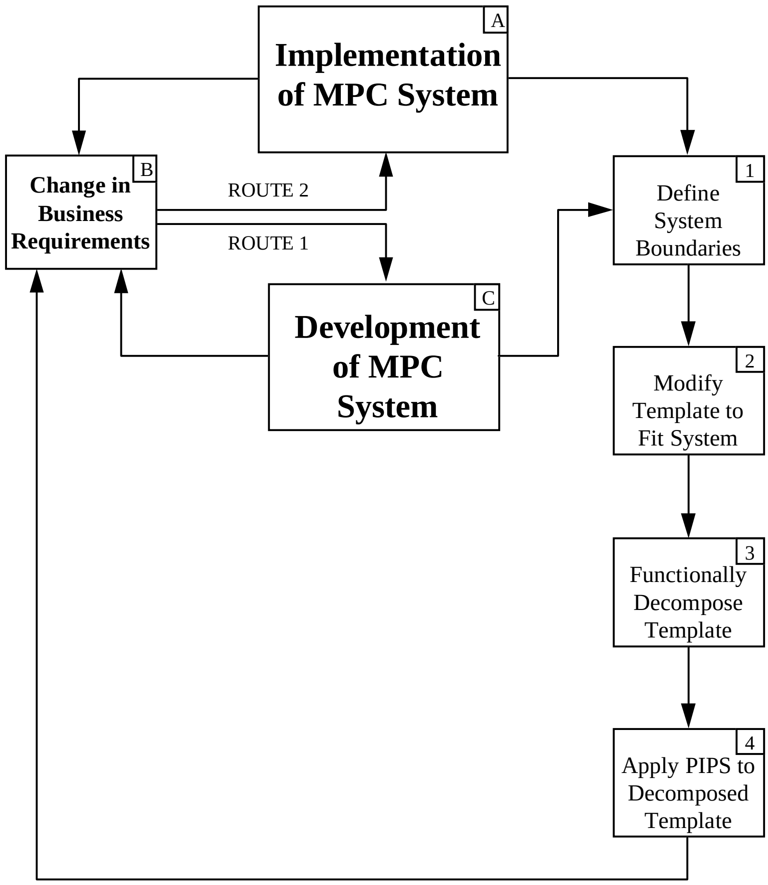


Figure 35 below details the process that should be followed through Box D.

Figure 35: The New Systems Development Life Cycle



The New Systems Development Life Cycle Methodology is based upon the New Systems Development Life Cycle diagram shown in Figure 35 above. This methodology consists of four stages, starting after a systems change and feeding back into the Systems Development Life Cycle. The four stages are:

1. Define the MPC system that is to be documented;
2. Modify the pre-defined template to fit the MPC system;
3. Decompose this modified template to find all levels of the MPC system;
4. Document each level of the MPC system using the PIPS documentation structure.

It can be seen that this methodology does not help the document creator select an appropriate technique as was the case in the Systems Development Life Cycle Methodology. This methodology helps the document creator create documentation by guiding them through a series of structured process stages. In rare occasions, where documentation highlights an error in the system the Systems Development Life Cycle Methodology should be used to correct system operation before re-starting the New Systems Development Life Cycle.

3.11.3.1 Define System Boundaries

This is the only part of the methodology that has not been covered in previous sections but is actually a crucial prerequisite for effective documentation. A key element of developing the documentation template is to initially define the system boundaries. In the same way as with parts of the Systems Development Life Cycle Methodology there is no simple formula that can be applied to identify the system. In some instances it will be easy to identify an MPC system, in other instances it will not.

For example, when defining the boundaries of a particular implementation of an MPC system the system boundaries could be defined as those business functions controlled by a particular database. This would appear to be a logical definition. There is a flaw in this definition, however. It is possible to control multiple manufacturing sites using one database. These sites can be in one physical location or globally spread. If, in a multiple site environment, centralised resources are implemented, (such as planning and sales), then these sites do in fact operate as one entity and should be considered to be part of one system. If these sites do not have central resources and operate as independent operating entities then they are not part of one system.

Ultimately, defining the system boundary is a matter of judgement. Companies are too diverse to define a prescriptive method for identification of boundaries. There are, however, a number of characteristics of independent and common entities that have been observed by the author:

1 **Separate Documentation**, (*each division or site is a different entity*):

- 1.1 Divisionalised or multi site company with multiple databases. Inter-divisional or inter-site trading is done using sales and purchase orders;
- 1.2 Divisionalised or multi site company with multiple databases. No trading between divisions or sites;
- 1.3 Divisionalised or multi site company with one database. Each division has separate facilities and has separate parts of the database. Inter-divisional or inter-site trading is done using sales and purchase orders;
- 1.4 Divisionalised or multi site company with one database. Each division has separate facilities and has separate parts of the database. No trading between divisions or sites.

2 **Common Documentation**, (*all divisions or sites are part of one big entity*):

2.1 Single site company with one database. All trading is done using sales and purchase orders;

2.2 Divisionalised or multi site company with one database. One or more facilities or parts of the database, (such as stores or sales), are shared between divisions or sites. Trading is done using sales and purchase orders or all demand is centralised.

The above list is not meant to be exhaustive but attempts to enable the reader to understand the process of defining the MPC system.

If the documentation exercise is undertaken immediately following the implementation of an MPC system then it is likely that the document creators will be very familiar with the system definition. In this case this part of the methodology will be extremely simple.

If the documentation exercise lags the implementation or is done by staff that were not involved in the implementation then the system definition may not be obvious and will need to be determined. Typically it will require interviews with personnel to compile this documentation.

Ferodo has a divisionalised structure with each division having its own database. As a result of this inter-divisional trading is done using sales and purchase orders and each division is responsible for its own MPC system documentation.

Once the system has been defined, the template should be applied.

3.11.3.2 Modify the Template to Fit the System

The template explained in section 3.11.1 and shown in Figure 32 has been derived from experience of a discrete batch environment and attempts to cover all aspects of manufacturing. This template may, however, contain functions that are not used in another discrete batch environment, (such as shop floor data capture), or may not match the manufacturing environment, (such as production lines without work order control).

For this reason the next stage of the methodology is to modify the template to reflect the functions used within the system being documented. The Chemicals Procurement example in section 3.11.2 illustrated this. This modified template is the basis of a full map of the system which can be derived by decomposing the modified template.

3.11.3.3 Functionally Decompose Template

Once the template has been modified it should be decomposed to determine all the functions, sub-functions and actions performed by each part of the system. This can be done in relatively simple cases by careful analysis of each part of the system. In complex cases IDEF0 models may need to be developed. For example the template within Ferodo would be decomposed to find that there is a manufacturing consumables part of the controlling consumables attribute of the stock control function. To control manufacturing consumables the following parts of the manufacturing database are used:

- Definition of parts;
- Definition of supplier items;
- Analysis of stocks of parts;
- Issuing purchase orders.

3.11.3.4 Apply PIPS to Decomposed Template

Once the first three steps of the methodology have been completed the documentation must be created. The PIPS levels of documentation must be provided to ensure that sufficient content is provided for the various reader types, (systems developers attempting to understand the system and system users attempting to understand the procedures).

The four PIPS levels are:

1. Purpose, (*the reasons for each function and action within the system*);
2. Interaction, (*the way each function and action interacts with others*);
3. Procedures, (*detailed operating instructions for each action*);
4. Software, (*detailed software manual for the database software*).

An IDEF0 model may have been created when defining and functionally decomposing the template to produce a system map. If so then this graphical representation could be used as the interaction documentation of the MPC system. Otherwise an IDEF0 model may be created for this purpose or verbose text used.

A detailed software manual will normally accompany a proprietary manufacturing database, (one of the justifications for the purchase of a system of this complexity). It may not be necessary, therefore, for the document creators to write this documentation. If satellite systems are used, (as shown in section 3.5.1), it is necessary to create software documentation.

If the documentation exercise identifies an error in the MPC system this stage of the methodology feeds back into the Systems Development Life Cycle Methodology.

3.11.4 Document Storage and Distribution

It can be seen that use of the New Systems Development Life Cycle Methodology will yield a significant amount of documentation, (though this is to be expected considering the complexity of MPC systems). This documentation should be created using a standard layout so that it can be understood throughout the organisation. Using a standard layout has the benefit of allowing the system users to create their own documentation once the system map has been defined.

Once all the documentation has been created a mechanism for its storage and distribution is necessary. The simplest mechanism is a paper based system as defined in section 2.16.2. There are a number of problems with this approach when a large number of documents need to be controlled, (as in section 2.16.3). A better mechanism is to use one of the two electronic documentation systems described in section 2.17. The approach preferred by the author, however, is to adopt a hybrid system combining the best parts of the 2 electronic approaches, as in Lotus Domino, as shown in section 3.10.

Lotus Domino allows documentation to be created and maintained in secure document databases. The contents of these databases can be linked together electronically using 'hypertext links' to allow easy navigation through them. In addition to this the databases can be configured with a number of indexes to allow easy identification of interesting documents and the system allows the databases to be searched for any documents containing particular keywords or keyphrases. Once these databases have been configured they can be accessed using a standard web browser to allow general, read-only, access to the documents, within the company or across the Internet.

3.12 General Applicability of the New Methodology

It has been shown that the MRPII database structure can be used to store data in the JIT and OPT environments as well as in the MRPII environment, (end of section 2.6, section 2.7 and section 3.6). This is because, despite there being differences between the three approaches to MPC, there is a similarity in their purpose, (to support and control the conversion of raw material into saleable product). There is also, therefore, a similarity in the documentation required to accompany these systems.

Section 3.6 showed that the Systems Development Life Cycle Methodology was equally applicable in all three MPC approaches. The methodology was developed into the New Systems Development Life Cycle Methodology, which draws requirements for documentation from the development of an MPC system. Part of the methodology is the top level template used to derive the map of the system. This template was derived from documentation of an MRPII package in a discrete batch manufacturing environment and needs to be modified each time it is used to document a system. The act of modifying this template as part of the methodology renders it generally applicable.

The New Systems Development Life Cycle Methodology, therefore, has the potential to be used to assist with creating documentation in an MRPII, JIT and OPT environment. The methodology is most applicable in a discrete batch environment but does, however, allow itself to be modified to suit any manufacturing environment.

Chapter 4 will discuss several applications of the Systems Development Life Cycle Methodology and the results of a series of structured interviews designed to determine the validity of the work presented in this thesis.

Chapter 4. Validation of the Systems Development Life Cycle Methodology

4.1 Introduction

The research objectives presented in Chapter 1 required that the research was validated. The literature review presented in Chapter 2 and the development of the methodologies presented in Chapter 3 showed that the case study company could be considered to provide a representative study but it was apparent to the author that the claimed broader applicability, whilst appearing to be true, should be tested.

Section 3.5.2 presented a series of analytical discussions supporting the Systems Development Life Cycle Methodology. The research objectives, however, required a more thorough validation. Both Action Research, (section 1.8) and the Scientific Method, (section 2.10), allow for research output to be tested. In Action Research, (the research methodology adopted by this research), if testing is to be done it is achieved by re-applying the research output to another similar business problem. The scientific method does not define an specific approach to adopt when testing research output. It was decided that two different validation techniques would be used:

- Questionnaire survey or a series of structured interviews;
- Practical application of the Systems Development Life Cycle Methodology.

Either questionnaire surveys or structured interviews could be used to help validate the Systems Development Life Cycle Methodology and its extension into the New Systems Development Life Cycle Methodology, (with Documentation). This chapter will undertake a miniature literature review, detailing the two approaches and will show that in this instance interview is the preferred option, (section 4.2).

An interview schedule was developed, along with a detailed agenda, (section 4.3) and a range of interviewees were selected, from different businesses and backgrounds. A series of interviews were carried out and the results were tabulated, (section 4.4). Following the interviews an analysis of the results was undertaken and conclusions drawn, (section 4.5).

The Systems Development Life Cycle Methodology was then tested. The author was able to work with three different companies, all of whom had system development requirements. The companies were all in different industries with different change requirements. The Systems Development Life Cycle Methodology was applied to systems development projects within the three companies and this chapter will end by presenting the results of these development projects, (section 4.6).

4.2 Interrogative Validation Techniques

There are a number of texts available, detailing the relative advantages and disadvantages of using a questionnaire as a fact finding technique as opposed to conducting structured interviews, (Tuckman 1978, James 1989, Caswell 1991, Waters, 1997). Both Waters and James state that of the two fact finding techniques, interview is the preferable. One of the reasons given is that interviews have a very high response rate. They also allow the interviewer to explain any unclear questions, (Waters 1997).

The same is not true of questionnaire as a fact finding technique. One major problem with questionnaires is that the response rates can be extremely low. Reynolds and Wainwright, (1997), present a survey on information integration which received a response rate of only 27%.

Baines *et al*, (1996), discuss two surveys, one of which, about the use of IDEF0, had a 20% response rate, the other, about modelling in manufacturing, had a response rate of only 6.3%. Hussain and Hussain, (1995), state that a response rate of between 15% and 35% could normally be expected from questionnaire surveys.

Another problem with questionnaire surveys is that questionnaires are extremely difficult to construct. Questionnaire respondents are isolated from the question setter. As a result, the question setter cannot be approached to help clarify any apparently ambiguous points. This leaves poorly constructed questionnaires open to misinterpretation, (Caswell 1991, Waters 1997).

A face-to-face interview allows the interviewer to visually check facial and body language clues given by the interviewees. They also generally give the interviewer a longer time to ask each question and probe for detail. They are, however, expensive as interviewers need to be employed and can result in bias if the interviewer is allowed to 'interpret' responses, (Caswell 1991, Waters 1997).

In summary, questionnaires are cheaper than interview and can have a wider sample, (though there will be a lower response rate). The bias of the interviewer is removed but bias is introduced in the form of respondents personal feelings to a situation.

Waters, (1997), illustrates an example of this where only people with a bad holiday experience respond to a questionnaire about holidays. Questionnaire also has the advantage of allowing respondents to spend longer deciding upon the 'best' answer to a question. This in itself is a problem as questionnaires take longer to complete and return, (Caswell 1991, Waters 1997).

It was decided that interview would be adopted as the technique to use in this situation. Interview was chosen for three main reasons:

1. Due to the complexity of the subject matter the questionnaire would require detailed explanation, possibly resulting in misunderstanding of the subject matter and people failing to respond due to the volume of material to read;
2. A high response rate was required;
3. The time available to conduct a validation exercise was limited.

James, (1989), advises that the questions to be asked during an interview should be pre-prepared in advance, defining clear objectives for the interview. Waters, (1997), also advises that a pilot interview should be conducted, if possible, to validate the interview questions before embarking on the main interview.

Tuckman, (1978), James, (1989) and Waters, (1997), all emphasise that the choice of question format is key to a successful interview. Waters advises that questions should be kept as simple as possible, asking positive, direct questions. Waters also advises that questions should be created that allow coding of the answers for analysis.

James defines the closed and open question as follows:

- A closed question solicits a choice of pre-defined responses, (such as Yes/No);
- An open question allows interviewees to give a verbose answer. James also suggests that open questions are more difficult to analyse.

Tuckman, (1978), goes further by showing that there are a number of different types of response that a question can request and suggests that the yes/no response, (or categorised response), is of interest, rather than, for example, the ranked response.

This is because the categorised response is the easiest to analyse after the interview as all interviewees have answered the same question from a selected list of answers.

These responses can then be analysed using a response counting analysis technique where, for example, the number of positive and negative answers are evaluated and conclusions drawn. An alternative approach would be to assign relative points to each answer and give a total score to each interview.

It was decided that during the interview closed questions would be asked, if possible, with categorised answers requested. Furthermore a standard set of questions would be asked in an attempt to ensure that the interviews all follow the same format. There would be two types of question, one, with a yes/no answer asked for, determining what the interviewee thinks of a particular part of the work presented in this thesis, broken down by section.

These questions would allow the interview to work towards the end of each section and the second type of question. The second type of question would be specifically designed to validate or disprove each section, with a choice of 'Validated', 'Modified', or 'Rejected'. The sections would be:

1. A breakdown of the construction of the Systems Development Life Cycle Methodology;
2. A breakdown of the structured questions;
3. A breakdown of the Indicative Ratios;
4. A breakdown of the extension of the methodology to include documentation;
5. A breakdown of the top level MPC systems documentation template;
6. A breakdown of the PIPS levels of documentation.

4.3 The Validation Interviews

An interview template was created from the rules detailed in section 4.2 above. The template starts with a number of reproductions from this thesis, allowing the author to outline the work done in this thesis. The reproductions that were presented to the interviewees during the course of the interview are as follows:

1. The Systems Development Life Cycle Methodology diagram, (from Figure 29);
2. The table of techniques available to systems developers, (from Figure 30);
3. The Decision Support Questions, (from section 3.5.3.1);
4. The decision support Indicative Ratios, (from section 3.5.3.2, Figure 31);
5. The Systems Development Life Cycle with Documentation, (from Figure 35);
6. PIPS, (from section 3.11.2);
7. The Top Level MPC Documentation Template, (from Figure 32).

After presenting the reproductions the interview template continues with a series of questions to be asked during the interview. The interview question sections were presented in section 4.2. The interview template is contained in the appendices as section 6.5. Each series of questions ends with a brief description of the intended purpose of the section from a data collection viewpoint to assist with gaining an understanding of the interview agenda.

Before fully embarking on the interviews a series of pilot interviews were undertaken to test the interview format. These interviews, (interviewees 1 and 2 from the results shown in section 4.4 below), validated the interview format but indicated that the theory behind the work presented should be explained in more detail before asking questions.

4.4 The Validation Interview Results

The interview agenda presented in section 4.3 and in the appendices as section 6.5 is broken down into a number of sections, each one relative to a particular point of the work presented in this thesis. This section of the thesis will present and analyse the outcome of each section of the interview in turn, breaking the sections down as follows:

1. Analysis of Systems Analysis / MRPII Experience;
2. Analysis of the Requirement for a formal systems development methodology;
3. Analysis of the Systems Development Life Cycle Methodology;
4. Analysis of the structured questions;
5. Analysis of the Indicative Ratios;
6. Analysis of the extension of the methodology to include documentation;
7. Analysis of the top level MPC systems documentation template;
8. Analysis of the PIPS levels of documentation.

The interviewees selected contained a mixture of people from a number of different companies. Some of the interviewees were experienced in systems analysis as applied to business control systems, the others were relatively inexperienced. All of the first group were familiar with MRPII systems. Most of the second group, whilst inexperienced in systems analysis, were experienced MRPII practitioners. The interviews were conducted over a three week period and the interview results presented in Figure 36 to Figure 43 below were collected.

The results of the interviews are presented below, organised by section. A combined percentage approval has been calculated and is presented at the bottom of each section.

Figure 36: Analysis of Systems Analysis / MRPII Experience

Interviewee Number	Job Description	Systems Analysis Experience (Yes/No)	MRPII Experience (Yes/No
1	Senior Consultant	Yes (20 Years)	Yes (13 Years)
2	Senior Consultant	Yes (12 Years)	Yes (12 Years)
3	Senior Consultant	Yes (15 Years)	Yes (15 Years)
4	MRP Systems Developer	Yes (4 Years)	Yes (4 Years)
5	Business Analyst	Yes (23 Years)	Yes (10 Years)
6	Logistics Manager	No (<1 Year)	Yes (20 Years)
7	IT Systems Manager	Yes (15 Years)	Yes (13 Years)
8	Development Manager	Yes (10 Years)	Yes (5 Years)
9	Financial Consultant	Yes (26 Years Periodic)	No
10	Business Analyst	Yes (10 Years)	Yes (7 Years)
11	Business Analyst	Yes (17 Years)	Yes (17 Years)
12	MRP Systems Developer	Yes (16 Years)	Yes (12 Years)
13	Senior Consultant	Yes (15 Years)	Yes (15 Years)
14	Consultant	Yes (15 Years)	Yes (15 Years)
15	Sales Director	Yes (12 Years)	Yes (12 Years)
16	Consultancy Director	Yes (12 Years)	Yes (12 Years)
17	Managing Director	Yes (20 Years)	Yes (20 Years)
18	Senior Developer	Yes (10 Years)	Yes (3 Years)
19	Consultant	Yes (20 Years)	Yes (16 Years)
20	Systems Developer	Yes (7 Years)	Yes (2 Years)

Interviewees 4, 6 and 9 could be considered to be relatively inexperienced.

Figure 37: Analysis of the Requirement for a Methodology

Interviewee Number	Would a methodology help with systems development (Yes/No)	Why?
1	Yes	
2	Yes	
3	Yes	
4	Yes	
5	Yes	
6	Yes	
7	Yes	
8	Yes	
9	Yes	
10	Yes	
11	Yes	
12	Yes	
13	Yes	
14	Yes	
15	Yes	
16	Yes	
17	Yes	
18	Yes	
19	Yes	
20	Yes	
Results	100% Yes	

Figure 38: Analysis of the Systems Development Life Cycle Methodology

Interviewee Number	Change Process before Modify Core (Yes/No)	Satellite System before Modify Core (Yes/No)	Sequence Appropriate (Yes/No)	Validated, Modified, Rejected	Why?
1	Yes	Yes	Yes	Validated	
2	Yes	Yes	Yes	Validated	
3	Yes	Yes	Yes	Validated	
4	Yes	Yes	Yes	Validated	
5	Yes	No	No	Modified	Mod. 1st
6	Yes	Yes	Yes	Validated	
7	Yes	Yes	Yes	Validated	
8	Yes	Yes	Yes	Validated	
9	Yes	Yes	Yes	Validated	
10	Yes	Yes	Yes	Validated	
11	Yes	Yes	Yes	Validated	
12	Yes	Yes	Yes	Validated	
13	Yes	Yes	Yes	Validated	
14	Yes	Yes	Yes	Validated	
15	Yes	Yes	Yes	Modified	Detail
16	Yes	Yes	Yes	Modified	Detail
17	Yes	Yes	No	Validated	Detail
18	Yes	Yes	Yes	Validated	
19	Yes	No	No	Validated	Detail
20	Yes	No	No	Modified	Mod. 1st
Results	100% Yes	85% Yes	80% Yes	Validated	

Figure 39: Analysis of the Structured Questions

Interviewee Number	Of use if not experienced (Yes/No)	Of use if some experience (Yes/No)	Of use if experienced (Yes/No)	Validated, Modified, Rejected	Why?
1	Yes	Yes	No	Validated	Lack of detail
2	No	Yes	Yes	Modified	Lack of detail
3	Yes	No	No	Validated	Use experience
4	No	Yes	No	Validated	Would ask for help
5	Yes	Yes	Yes	Validated	
6	Yes	Yes	No	Validated	Use experience
7	Yes	Yes	Yes	Validated	
8	Yes	Yes	No	Validated	Done naturally
9	Yes	Yes	No	Validated	Use experience
10	Yes	Yes	Yes	Validated	
11	Yes	Yes	Yes	Validated	
12	Yes	No	No	Validated	Use experience
13	No	Yes	Yes	Modified	Lack of detail
14	Yes	No	No	Validated	Use experience
15	Yes	Yes	Yes	Validated	
16	Yes	Yes	No	Validated	Done Naturally
17	Yes	Yes	Yes	Validated	
18	Yes	Yes	Yes	Validated	
19	Yes	Yes	Yes	Validated	
20	Yes	Yes	Yes	Validated	
Results	85% Yes	85% Yes	55% Yes	Validated	

Figure 40: Analysis of the Indicative Ratios

Interviewee Number	Of use if not experienced (Yes/No)	Of use if had some experience (Yes/No)	Of use if experienced (Yes/No)	Ratios Realistic (Yes/No)	Validated, Modified, Rejected	Why?
1	No	No	No	Yes	Rejected	Maths
2	Yes	Yes	Yes	Yes	Validated	
3	No	No	No	No	Rejected	Complex
4	No	No	Yes		Modified	Complex
5	No	No	Yes	Yes	Rejected	Complex
6	Yes	Yes	Yes		Validated	
7	Yes	Yes	Yes	Yes	Validated	
8	No	No	No	Yes	Rejected	Specific
9	No	No	Yes		Rejected	Complex
10	Yes	Yes	Yes	Yes	Validated	
11	Yes	Yes	Yes	Yes	Validated	
12	No	No	No	Yes	Rejected	Complex
13	Yes	Yes	Yes	Yes	Validated	
14	Yes	No	No	No	Rejected	Complex
15	Yes	Yes	Yes	Yes	Validated	
16	Yes	Yes	Yes	Yes	Validated	
17	Yes	Yes	Yes	Yes	Validated	
18	No	No	No	Yes	Modified	Simple
19	Yes	Yes	Yes	Yes	Validated	
20	No	No	No	Yes	Rejected	Specific
Results	55% Yes	50% Yes	65% Yes	88% Yes	Validated	

Figure 41: Analysis of the Methodology Extension to include Documentation

Interviewee Number	Documentation part of systems development (Yes/No)	Sequence Appropriate (Yes/No)	Validated, Modified, Rejected	Why?
1	Yes	Yes	Validated	
2	Yes	Yes	Validated	
3	Yes	No	Modified	Difficult to support
4	Yes	Yes	Validated	
5	Yes	Yes	Modified	Too detailed
6	Yes	Yes	Validated	
7	Yes	Yes	Validated	
8	Yes	Yes	Validated	
9	Yes	Yes	Validated	
10	Yes	Yes	Validated	
11	Yes	Yes	Validated	
12	Yes	Yes	Validated	
13	Yes	Yes	Validated	
14	Yes	Yes	Validated	
15	Yes	Yes	Validated	
16	Yes	Yes	Validated	
17	Yes	Yes	Modified	Too detailed
18	Yes	Yes	Validated	
19	Yes	No	Modified	Lacks Validation Stage
20	Yes	Yes	Validated	
Results	100% Yes	90% Yes	Validated	

Figure 42: Analysis of the Top Level MPC Systems Documentation Template

Interviewee Number	Of use if not experienced (Yes/No)	Of use if some experience (Yes/No)	Of use if experienced (Yes/No)	Validated, Modified, Rejected	Why?
1	Yes	Yes	Yes	Validated	
2	Yes	Yes	Yes	Validated	
3	Yes	Yes	Yes	Validated	
4	Yes	Yes	Yes	Validated	
5	Yes	Yes	Yes	Modified	Too Detailed
6	Yes	Yes	Yes	Validated	
7	No	No	No	Rejected	Irrelevant
8	Yes	Yes	Yes	Validated	
9	Yes	Yes	Yes	Validated	
10	No	Yes	Yes	Validated	Too Detailed
11	No	Yes	No	Modified	Assumes Prior Knowledge
12	Yes	Yes	Yes	Validated	
13	Yes	Yes	Yes	Validated	
14	Yes	Yes	Yes	Validated	
15	Yes	Yes	Yes	Validated	
16	Yes	Yes	Yes	Validated	
17	Yes	Yes	Yes	Validated	
18	Yes	Yes	Yes	Validated	
19	Yes	Yes	Yes	Validated	
20	Yes	Yes	Yes	Validated	
Results	85% Yes	95% Yes	90% Yes	Validated	

Figure 43: Analysis of the PIPS Levels of Documentation

Interviewee Number	Of use if not experienced (Yes/No)	Of use if some experience (Yes/No)	Of use if experienced (Yes/No)	Validated, Modified, Rejected	Why?
1	Yes	No	No	Modified	Lack of detail
2	Yes	Yes	Yes	Validated	
3	Yes	Yes	Yes	Validated	
4	Yes	Yes	Yes	Validated	
5	Yes	Yes	Yes	Modified	
6	Yes	Yes	Yes	Validated	
7	Yes	Yes	Yes	Validated	
8	Yes	Yes	Yes	Validated	
9	Yes	No	No	Modified	Done Naturally
10	Yes	Yes	Yes	Validated	
11	Yes	Yes	Yes	Validated	
12	Yes	No	No	Validated	Done Naturally
13	Yes	Yes	Yes	Validated	
14	Yes	Yes	Yes	Validated	
15	Yes	Yes	No	Modified	Done Naturally
16	Yes	Yes	No	Modified	Done Naturally
17	Yes	Yes	No	Modified	Done Naturally
18	Yes	Yes	Yes	Validated	
19	Yes	Yes	Yes	Validated	
20	Yes	Yes	Yes	Validated	
Results	100% Yes	85% Yes	70% Yes	Validated	

4.5 The Validation Interview Findings

The results of the interviews, presented in section 4.4, can be summarised as in Figure 44 below. The first observation that can be made is that there is no significant differential between the results gained from interviews with experienced analysts and the results of interviews with inexperienced analysts. The results have, therefore, been sorted by question and conclusions will be drawn from the percentage of positive answers received to each question and the most common answer to the question – 'Validated, Modified, Rejected', asked at the end of each section.

Figure 44: Results of Validation Interviews

Question	**% Of Positive Answers**	**Validated, Modified, Rejected**	**Why?**	**Average % of +ve Answers**
Systems Development Life Cycle Methodology				
Would a Methodology help with Systems Development	100	Validated	One wanted core modified before satellite systems investigated Too Detailed	91.25
Alter Process before Mod. Core	100			
Sat. Systems before Mod. Core	85			
Sequence Appropriate	80			
Decision Support Questions				
Of use if not experienced	85	Validated	Lack of detail Precludes experience	75
Of use if some experience	85			
Of use if experienced	55			
Indicative Ratios				
Of use if not experienced	55	Validated	Too Complex Too Specific Too Simple Dislike Maths	64.5
Of use if some experience	50			
Of use if experienced	65			
Ratios Realistic	88			
New Systems Development Life Cycle Methodology, (with Documentation)				
Is Documentation part of systems development	100	Validated	Difficult to Support Too Detailed Validation Stage	95
Sequence Appropriate	90			
Top Level MPC Systems Documentation Template				
Of use if not experienced	85	Validated	Too Detailed Irrelevant Assumes Prior Knowledge	90
Of use if some experience	95			
Of use if experienced	90			
PIPS				
Of use if not experienced	100	Validated	Lack of detail Done naturally	85
Of use if some experience	85			
Of use if experienced	70			

It can be seen from Figure 44 that every interview section has been validated and that all the sections receive an average approval percentage above 60%.

The results presented in Figure 44 can be broken down into two groupings:

1. The Methodologies at a top level:

   The Systems Development Life Cycle Methodology and the New Systems Development Life Cycle Methodology, (with Documentation).

2. Sub-Sections of the Methodologies:

   Decision Support Questions; the Indicative Ratios; the Top Level MPC Systems Documentation Template and PIPS.

Both of the methodologies receive an average approval percentage above 90% indicating that the vast majority of interviewees interviewed approve of the sequence, validity and requirement for the methodologies.

The responses received for the sub-sections of the methodologies are more varied, however. The interviewees suggested that the Decision Support Questions and PIPS would be of more use to inexperienced analysts. When asked why the interviewees explained that they thought that these approaches would be applied naturally by experienced analysts. The opposite is true of the Indicative Ratios and Top Level MPC Systems Documentation Template. Interviewees believed that these would be of more use to experienced analysts, as a certain level of experience would be needed to understand them but that they would then be useful tools to check decisions being made.

Section 4.6, next, will discuss the application of the Systems Development Life Cycle Methodology in three further systems development projects undertaken by the author.

4.6 Application of the Systems Development Life Cycle Methodology

Following development of the Systems Development Life Cycle Methodology and the decision support tools, additional systems development work was undertaken in three further companies. The purpose of this work was to confirm that the methodology had application outside the main collaborating company. This section will describe how the Systems Development Life Cycle Methodology and the decision support tools were applied in these cases.

The companies were chosen for inclusion in the research for three reasons; firstly, they were in the process of developing their systems and therefore, the application of the methodology was pertinent. Secondly, the companies were in different market sectors to the collaborating company, Ferodo, and therefore, provided additional breadth to the research. Finally, all were users of the MFG/PRO ERP package and the familiarity of the author with this software enabled access to be gained readily. For commercial reasons, it is not possible to discuss the specifics of these projects and therefore, they shall be referred to as case study companies A, B and C.

Case Study Company A

Case study company A is a distributor of clinical equipment. They decided to upgrade to the current release of MFG/PRO, but realised that it would not fully satisfy business requirements in two areas of the service and support management module;

- Scheduling of preventative maintenance visits for product covered under contract;
- Enhanced validation and data entry for the creation of invoices for support work.

Case Study Company B

Case study company B, a medical equipment manufacturer and service provider, was also upgrading to the current release of MFG/PRO. Following an analysis of the new software, it was discovered that there were two system requirements that were not satisfied by the standard package. These two change requirements were also in the service and support management module;

- Support for a variety of service contract types offered by the company;
- Enhanced validation and data entry for the creation of invoices for support work.

Case Study Company C

In response to customer demand case study company C, a supplier of MFG/PRO, needed to provide a system for mobile data collection. These systems are typically used by maintenance technicians when supporting products in the field. In response to customer demand, case study company C was required to allow these technicians to automatically exchange data, (using mobile telecommunication networks), with MFG/PRO, using appropriate hardware and software. There was a need to modify MFG/PRO to support this requirement.

The system needed to allow office based call-centre staff to automatically transmit information logged in MFG/PRO against a customer call to a technician. Once the technician received this information, the system needed to allow the technician to send back periodical updates. Office staff could then access the updated information. The system also needed to allow users to review a log of the information exchanged.

4.6.1 Case Study Company A

Analysis of the change requirements at case study company A showed that they were fundamental to the operation of MFG/PRO's service and support management module. The change requirements were that each time a maintenance contract was raised on MFG/PRO, it would be necessary to configure an individual preventative maintenance pattern. Similarly, each time a service call invoice was raised through the system, extra data needed to be stored and some new data processing needed to occur. Initial application of the Systems Development Life Cycle Methodology suggested that one of the integration techniques should be applied.

The service and support management module has three main functions. These are contract creation, service call creation and service call invoicing. The change requirements fundamentally affected two of these three functions. With reference to the decision support Indicative Ratios accompanying Figure 31 it could be seen that for each change the ratio of the number of transactions within the core system to the number of transactions in the auxiliary system was approximately 3:1, (i.e. changes were needed to one function out of three), and therefore, option 3 of the Indicative Ratios was further investigated.

Whilst the change requirements were for two out of the three core areas of the service and support management module, the new system required a significant amount of new data to be created and updated within MFG/PRO. This high transaction rate between systems indicated that either option 3.5 or option 3.6 of the decision support Indicative Ratios accompanying Figure 31 should be applied in both cases. The Decision Support Questions were then applied in both cases as summarised in Figure 45.

Figure 45: Decision Support Questions applied to Case Study Co. A

Should the Process be Changed?		**Y/N**
1	Is it possible to change the process to meet the changed business requirements, (taking into account external accreditation and quality guidelines)? **Rationale:** the marketplace defined the business process and therefore, there was little opportunity for change.	No
Can it use an Available Module?		
1	Is there a module available? **Rationale:** no appropriate software was available from the ERP software vendor.	No
Can it use Spare Fields in the Core?		
1	Are there any applicable spare fields in the core, (that store data in the same format, preferably using the same indexes)? **Rationale:** the requirement was far to complex to be satisfied by this approach. As discussed in section 3.5.1.1, use of spare fields can generally only be used to address relatively simple requirements.	No
Can it use a Satellite System?		
1	Can an auxiliary database be written to meet business requirements, (remembering the link into the existing systems)? **Rationale:** the data interchange requirements between the core systems and an auxiliary system would be highly complex.	No
Must it Modify the Core?		
1	Can modifications be made to the core to meet the business needs?	Yes
2	Is there sufficient programming resource available to do the work?	Yes
3	Can the work be done within budget, (remembering the continuing cost of each upgrade)?	Yes

The changes required additional functionality not available in an additional module and the complexity of the changes ruled out the use of spare fields. The number of changes required within the core system meant that it would be extremely difficult to design a satellite system and as a result the modification approach was selected for both changes.

A series of modifications were designed by the author and an implementation project team. An independent programming company were commissioned to program the changes to the core system. The changes have been successfully applied and the new functionality is in use.

4.6.2 Case Study Company B

The two change requirements at case study company B were potentially quite different. In the first instance they needed to be able to process two fundamentally different types of service contract through MFG/PRO. Existing functionality fully catered for one of the two types of contract, the other contract type, however, did not fit into the standard functionality. In the second instance the standard mechanism for service call invoicing did not have the required data fields or functionality. Initial investigation of the Systems Development Life Cycle Methodology also suggested that one of the integration techniques should be applied in both instances.

Case study companies A and B are similar in that they use the same three functions within their service and support management modules; contract creation, service call creation and service call billing. The change requirements also fundamentally affected two of these three functions. Option 3 of the decision support Indicative Ratios was further investigated. In the first instance, the two types of contracts were split on a roughly even basis. The new types of contract would be mainly self-contained, with very few transactions moving between the core and the auxiliary system, the ratio was estimated at approximately 1:1, (i.e. both contract types would be transacted at approximately the same rate), and as a result option 3.5 was selected. The Decision Support Questions were used to test this conclusion, (see Figure 46 below).

Figure 46: Decision Support Questions applied to Case Study Co. B – Part 1

Should the Process be Changed?		**Y/N**
1	Is it possible to change the process to meet the changed business requirements, (taking into account external accreditation and quality guidelines)? **Rationale:** handling the new contract type was a radically new requirement and could not be addressed by a modification of existing practices.	No
Can it use an Available Module?		
1	Is there a module available? **Rationale:** no appropriate software was available from the ERP software vendor.	No
Can it use Spare Fields in the Core?		
1	Are there any applicable spare fields in the core, (that store data in the same format, preferably using the same indexes)? **Rationale:** the requirement was far too complex to be satisfied by this approach.	No
Can it use a Satellite System?		
1	Can an auxiliary database be written to meet business requirements, (remembering the link into the existing systems)?	Yes
2	Is there sufficient programming resource available to do the work?	Yes
3	Can the work be done within budget?	Yes

The new contracts system was required to operate in conjunction with the existing system and the functionality was not available in any other module of MFG/PRO. The scale of the changes was such that spare fields could not be used. The use of satellite systems was investigated. The standard contracts programs existed as a self-contained series of programs. On investigation it was seen that it would be possible to copy the existing programs and modify them, using them as a template for the creation of a satellite system for the new contracts. A satellite system was designed by the author and the programming work was undertaken by the MFG/PRO vendor. The new contracts system has been successfully implemented.

The invoicing changes were similar, in principle, to those found in case study company A. Analysis of the decision support Indicative Ratios revealed the same result, that either option 3.5 or 3.6 should be selected. Figure 47 presents the results of the Decision Support Questions.

Figure 47: Decision Support Questions applied to Case Study Co. B – Part 2

Should the Process be Changed?		**Y/N**
1	Is it possible to change the process to meet the changed business requirements, (taking into account external accreditation and quality guidelines)? **Rationale:** this was a new requirement that could not be addressed through a modification of existing practices.	No
Can it use an Available Module?		
1	Is there a module available? **Rationale:** no appropriate software was available from the ERP software vendor.	No
Can it use Spare Fields in the Core?		
1	Are there any applicable spare fields in the core, (that store data in the same format, preferably using the same indexes)? **Rationale:** the requirement was far too complex to be satisfied by this approach.	No
Can it use a Satellite System?		
1	Can an auxiliary database be written to meet business requirements, (remembering the link into the existing systems)? **Rationale:** the interchange of data between an auxiliary database and the core system would be highly complex.	No
Must it Modify the Core?		
1	Can modifications be made to the core to meet the business needs?	Yes
2	Is there sufficient programming resource available to do the work?	Yes
3	Can the work be done within budget, (remembering the continuing cost of each upgrade)?	Yes

There was no additional module available to satisfy the invoicing system changes. Though there were some spare fields available in the standard invoicing system, these were inadequate for the scale of the changes required. It was determined that it would not be practical to design a satellite system to satisfy the change requirements and as a result the modification approach was chosen.

The modifications were designed by the author and the programming was undertaken by the MFG/PRO vendor. The invoicing changes have been applied and are being used successfully.

4.6.3 Case Study Company C

In a service environment it is common for service technicians to be scheduled and controlled remotely. Mobile telephones and pagers have been used to provide this cover. Despite the decreasing cost of mobile telephones, they are still expensive to provide and maintain. Pagers are a cheap alternative, but only allow unidirectional communication. Case study company C needed to develop a mobile data communications system, (a two way text messaging solution), to work alongside MFG/PRO. The existing system contained paging functionality, but did not support bi-directional messaging. The expectation was that the new system could be designed in such a way as to be used by a number of the customers of case study company C.

Standard mobile data communications systems require three components:

1. Host system data processing and dispersal, (i.e. an ERP database);
2. A communications network, (transmitters, cabling, control computers etc);
3. A hand-held data processing and display unit, (i.e. palm-top personal computer).

Case study company C were not able to provide a communications network or the hand-held data processing units. Analysis of the Systems Development Life Cycle Methodology indicated that the best option available was to use an independent package, linked to MFG/PRO. There were a number of these solutions available, all of which required data to be imported and exported through a standard interface.

The decision support Indicative Ratios are only used when choosing an integration technique and therefore, were not applied. The results of the Decision Support Questions are presented in Figure 48.

Figure 48: Decision Support Questions applied to Mobile Data – Part 1

Should the Process be Changed?		**Y/N**
1	Is it possible to change the process to meet the changed business requirements, (taking into account external accreditation and quality guidelines)? **Rationale:** the new requirement was based on the application of a new communications technological and therefore, changes in business processes were inapplicable.	No
Can it use an Available Module?		
1	Is there a module available? **Rationale:** the software was being written by an ERP software vendor as a suitable module did not exist in the existing ERP system.	No
Can it use Spare Fields in the Core?		
1	Are there any applicable spare fields in the core, (that store data in the same format, preferably using the same indexes)? **Rationale:** this was too complex a requirement to be satisfied by the use of spare fields.	No

Figure 48 continues on the next page.

Figure 48 Continued: Decision Support Questions applied to Mobile Data – Part 1

Can it use a Satellite System?		
1	Can an auxiliary database be written to meet business requirements, (remembering the link into the existing systems)? **Rationale:** the case study company did not possess the necessary skills or technology to implement a satellite system.	No
Must it Modify the Core?		
1	Can modifications be made to the core to meet the business needs? **Rationale:** the case study company did not possess the necessary skills to implement a satellite system.	No
Can it be an Independent Package?		
1	Is there an independent package available?	Yes
2	Does the independent package contain sufficient functionality?	Yes
3	Is the independent package affordable?	Yes
4	In the event of a linked system can the independent package link to the core?	Yes
5	Can this be achieved at an affordable cost?	Yes

In this instance it was relatively easy to apply the Systems Development Life Cycle Methodology as the company did not possess the skills needed to meet the requirement. The decision support tools were also not necessary. Whilst an additional package would provide most of the functionality required, it was evident that an interface to the mobile data system would need to be designed and written into MFG/PRO.

It was decided that a generic interface would be written in order to allow any of the mobile data solutions to be linked to the core systems to give flexibility to customers when configuring their systems (therefore, maximising sales). Upon investigation it was evident that the mobile data solutions all worked on the basis of transmissions of pre-defined ‘packets’ of data.

Any interface to the core systems would need to present data in pre-defined formats containing key data, which included some data that was not readily available in the existing product. A data extraction program was written to collect and format the data for transmission to the mobile data package. This program had no effect on the operation of the core and as such falls, outside of the scope of the Systems Development Life Cycle Methodology.

It was necessary to determine how to include the extra data for the interface within the core system and how to notify the mobile data interface which particular communications network provider had been chosen by the customer. The extra data that needed to be stored was mainly static, user identification codes and transmission protocols. The only dynamic data requirement was for a transmission log to be stored and for a status field to be updated.

Initial inspection of the Systems Development Life Cycle Methodology indicated that one of the integration approaches should be adopted and that incorporation with the core might be the preferred approach.

As most of the changes that needed to be made were to static data, the ratio of the number of transactions in the core system to the number of transactions in the auxiliary system was large, (as data maintenance was only done occasionally, when configuring the system), and as a result the decision support Indicative Ratios also suggested that incorporation with the core would be the best approach.

Figure 49 presents the results of the application of the Decision Support Questions to the mobile data interface.

Figure 49: Decision Support Questions applied to Mobile Data – Part 2

Should the Process be Changed?		**Y/N**
1	Is it possible to change the process to meet the changed business requirements, (taking into account external accreditation and quality guidelines)? **Rationale:** the new requirement was based on the application of a new communications technological and therefore, changes in business processes were inapplicable.	No
Can it use an Available Module?		
1	Is there a module available? **Rationale:** the software was being written by an ERP software vendor as a suitable module did not exist in the existing ERP system.	No
Can it use Spare Fields in the Core?		
1	Are there any applicable spare fields in the core, (that store data in the same format, preferably using the same indexes)?	Yes
2	If so, are there enough of these spare fields to use for the new system?	Yes

The author designed the mobile data interface using spare fields from within the core system. A free text field within the call creation program was used to store the transmission log data and the database was directly updated whenever a different status was detected. The mobile data interface was written by a programmer in the case study company and has now been successfully applied.

The system design projects were managed as follows:

- Case study company A;

  The author, a member of the commissioning company and a designer from the subcontract development company were responsible for the design effort which lasted approximately 2 weeks. The programming work and subsequent testing was handled by a team of programmers from within the subcontract development company and lasted approximately 2 months.

- Case study company B;

  The author had sole responsibility for the design effort and reported into company representatives for approval. Due to the scale of the changes this lasted approximately 2 months. The programming work and testing was mainly performed by one programmer, with occasionally assistance and took approximately 5 months.

- Case study company C.

  The author had sole responsibility for the design effort and consulted with a potential user to assist with decision making. This lasted approximately 1 month. The programming and testing was handled by one programmer and lasted approximately 2 months.

One of the disadvantages of industrial research is the difficulty of validating research findings. Philosophers such as Kuhn and Popper (Urmson and Ree 1993, Microsoft Encarta 1997) have challenged the simple interpretation of the scientific method. Popper argued that validation in an absolute sense in not possible, that researchers are not able to absolutely prove a scientific argument but only to provide evidence for its validity. Popper states that proof can only be offered to disprove scientific arguments.

Two strategies have been applied to help validate this research. Firstly by consulting a cross section of professionals who could reasonably be expected to be able to criticise the work. This provided evidence that the work is reasonable and coherent. Secondly, successful utilisation of the research outcome in practice in different industries. This provided evidence that the work has practical, (and general), utility.

Chapter 5 will end this thesis by drawing discussing the work presented, drawing conclusions and making recommendations for further work.

Chapter 5. Summary and Conclusions

This chapter will summarise the key points made in the main text of this thesis and will illustrate that the work has followed the Action Research cycle shown in Figure 4 on page 15, (section 5.1). The chapter will then draw conclusions before making recommendations for further work, (sections 5.2 and 5.3).

5.1 Summary

The work presented in this thesis has been split between two interrelated themes:

1. The Structured Development of Manufacturing Planning and Control Systems;
2. The Structured Documentation of Manufacturing Planning and Control Systems.

5.1.1 The Structured Development of MPC Systems

Chapter 2 compared and contrasted three approaches to Manufacturing Planning and Control; MRPII, JIT and OPT. It was shown that MRPII is based on a mature database structure which allows the storage of both static and dynamic manufacturing data. It was also shown that effective use of MRPII systems does not require that all parts of an MRPII database are used. Sections 2.6 and 2.7 showed that though the three approaches differ in their detail they have common data requirements and can use similar database structures, demonstrating the general applicability of the MRPII database.

Chapter 2 also discussed the development of methodologies and system modelling techniques. It was shown in section 2.10 that a methodology is a structured collection of tools and techniques to enable a perceived problem to be effectively tackled. An understanding of the composition of these methodologies and systems modelling techniques allowed derivation of the Systems Development Life Cycle Methodology.

Chapter 3 presented the work undertaken in systems development. The starting point was the hybrid approach to integrating new applications into core MPC databases, an approach which had been developed in prior research. This approach was adopted when developing some new systems within the collaborating company. The systems were all integrated into the pre-existing core MRPII database and were presented in section 3.4. They can be summarised as follows:

1. The Quality Analysis System;

   This system allowed the users to record and analyse manufacturing data.

2. The Property Recording System;

   This system allowed the users to record data from quality tests. The system subjected this data to Statistical Process Control analysis and presented the results.

3. The Manufacturing Instructions System;

   This system was designed to allow the users to store static product related data. The data stored was the key data used to define the manufacturing processes followed.

4. The Batch Tracking System.

   This system used spare fields available within the Quality Analysis System to store batch identification data. This data could then be used to identify the final destination of key components.

Reflecting on the experience gained whilst these systems were being developed within the collaborating company the author refined the hybrid approach to systems development and defined the 'satellite systems' approach to systems development, (see section 3.5.1). The new approach relies upon the definition of two key components of the new system; the core and the satellite. One of the differences between the hybrid approach and the satellite systems approach is the definition of the core system.

In the satellite systems approach the core is defined as the existing system structure, not necessarily just the pre-existing MRPII package. Change requirements are addressed by integrating newly developed 'satellite' systems with the most appropriate element of the current systems infrastructure. A key element of the satellite systems approach is that the new satellite systems should not directly update the core system, only access data from within the core in order to safeguard the core system.

Two types of satellite system were defined in order to help with development of these satellite systems. The satellite systems that could be used are as below:

1. Integrated with the core, (*Property Recording and Quality Analysis*);

   This type of system is integrated by creating two way data transfer links between the core system and the satellite system. The satellite system is accessed through the core system and uses data from the core when performing analyses.

2. Joined with the core, (*Manufacturing Instructions*).

   This type of system is integrated with a one way link to the core. The satellite system is accessed through the core but does not use data from within the core.

It should be noted that the satellite systems approach was defined as a result of developing the MRPII system within a discrete batch manufacturing company. Analysis of a number of manufacturing companies showed that the satellite systems developed within Ferodo could be used to complement the MPC systems they use or planned to use. These companies were predominantly using or planning to use MRPII systems themselves and this observation is, therefore, restricted to this subset of company.

The companies visited were listed and case studied in section 3.2.

In section 3.5.1 it was also shown that there are a number of different solutions that a systems analyst could adopt to solve a change in business requirements in these companies, rather than applying the satellite systems approach. These approaches are:

- Re-implementing a new core MRPII system;
- Modifying the business processes that necessitated the change;
- Purchasing an additional module for the existing core MRPII system;
- Using spare fields within the existing systems;
- Modifying the core systems;
- Purchasing an independent package.

Due to the variety of options available to systems analysts it was observed that there was a need for a methodology to guide a systems analyst through the process of choosing a solution. The Systems Development Life Cycle Methodology was developed for this purpose and was presented in section 3.5.2 as Figure 29 on page 178.

The methodology prioritises and details the choices open to systems analysts, placing the satellite systems approach to systems development fourth, (as shown in Figure 29). It is worth noting that the first option advocated by the methodology is to investigate whether a systems solution should be sought at all, as it was observed within the test environment that many 'systems problems' can be addressed by a change to the process.

An example of this happening, the Mix Batch Numbering system, was presented in section 3.4.4. The batch stamping technique in place did not support large numbers and rather than develop a better stamping technique a systems solution was applied. This added complexity to the systems and in the author's opinion, should have been avoided.

It is possible, with the benefit of hindsight, to see that the Property Recording system was designed in accordance with the methodology. The initial requirement included a graphical interface, which was causing significant design problems. This requirement was removed, enabling a satellite system to be designed. This is because it was recognised that it was mainly a cosmetic requirement. The same was true of the Manufacturing Instructions system.

Evidence has also been presented that shows that in the test environment modifying the operation of the core MRPII systems should only be considered if no alternative exists as this leads to a high degree of complexity when upgrading and supporting the systems.

Two decision support tools were presented in section 3.5.3 to assist both experienced and inexperienced systems analysts apply the methodology. These tools were:

1. The Decision Support Questions;

   A series of questions to be asked when determining which option to apply. A negative answer to any question leads the analyst to move onto the next option.

2. Indicative Ratios.

   A mathematical analysis of the problem which delivers ratios that can be used to determine which approach to apply.

The development of the Systems Development Life Cycle Methodology has followed the Action Research methodology. The initial hypothesis was that it would be possible to develop a methodology to assist with systems development. Work was done at Ferodo to help understand how satellite systems should be designed and implemented. Finally, reflecting on the work undertaken and literature allowed derivation of the Systems Development Life Cycle Methodology.

5.1.2 The Structured Documentation of MPC Systems

Section 3.8 continued the work on systems development by including work undertaken on the documentation of MPC systems. Initially system operation documentation was produced by the author as a familiarisation exercise. This took a period of three months to complete, from an extensive series of interviews. There were a number of assumptions made to restrict the content of the documentation to allow it to be condensed to a manageable size, as the publishing medium used was paper. This was explained in section 3.9.

The complexity of the documentation exercise identified a genuine business need and a solution to the document creation and distribution problem was sought. Two approaches to electronic documentation were presented in section 2.17. The two approaches were; Document Management systems that allowed detailed control over document storage and retrieval, (push systems) and Intranets, which have no control over document storage but provide very easy access to documentation using a web browser. It was decided that the two technologies should be combined and a product called Lotus Domino was used to achieve this. Using Lotus Domino an Intranet was developed by the author and the Intranet was present in section 3.10.

Despite allowing detailed control over document storage and easy access to read documents the Ferodo Intranet did not meet all the requirements as there was no guarantee that the information contained within the documentation was complete. It was shown in section 3.11 that a top level MPC system template would have assisted with the identification of documentation topics when creating the Ferodo Intranet.

For this reason a template was created, using analysis of the Ferodo MRPII package, experience of other MPC systems and a theoretical analysis of MRPII databases. The template was presented in section 3.11.1 as Figure 32 on page 214.

The analysis of different companies systems presented in section 3.2 showed that whilst the companies were similar they all differed in the detailed application of their systems. For this reason the template was defined at a top level, from experience of discrete batch manufacturing companies using MRPII databases and, therefore, represents a developed best fit within this type of industry. The template should not be considered to be an ideal or prescriptive MPC system documentation layout. Furthermore it is important that when applying the template the analysts analyse the template and adapt it to fit the specific business being documented, as no two MPC systems are alike.

It was also observed that a tool was required to convert this top level MPC template into comprehensive documentation. For this reason the PIPS four levels of documentation technique was derived from literature, as shown in section 3.11.2. It was shown that the PIPS levels of documentation allow document creators to provide adequate documentation to satisfy two types of readers within the companies studied:

1. Readers trying to understand how to perform specific tasks, (*emulating the minimalist approach to documentation*);
2. Readers trying to understand the operation of the system, (*emulating the elaboration approach to documentation*).

This is done by creating documents that define the purpose of the system, the interaction between different functions within a system, the detailed procedures to be followed by users of the system and the detailed operation of the software.

It was observed at the collaborating company that the implementation of new systems or the modification of the existing systems influence documentation. Following from this, it was suggested that a documentation exercise can highlight problems within an MPC system. If these problems are then used to initiate a change to the MPC system a feedback loop can be added from systems documentation to systems development.

It was recognised that a methodology was required to guide the documentation creator through the act of creating documentation using these techniques.

The New Systems Development Life Cycle Methodology, (with documentation), was created for this purpose and is presented in section 3.11.3 as Figure 35 on page 223. The New Systems Development Life Cycle Methodology includes a four stage procedure for the creation of MPC system documentation and is actually an extension of the Systems Development Life Cycle Methodology, to incorporate documentation. An Intranet was advocated as the document storage and distribution medium.

The development of the New Systems Development Life Cycle Methodology has also followed the Action Research methodology. The original paper based documentation exercise showed a need for a better approach to documentation and prompted this theme in the research. Following the identification of this need an Intranet was developed. Using Ferodo as the problem situation, the Intranet was developed and implemented using the Lotus Domino product. Reflection on the work undertaken allowed derivation of the Top Level MPC System Documentation Template and the PIPS approach which in turn led to the development of the New Systems Development Life Cycle Methodology from the Systems Development Life Cycle Methodology.

Chapter 4 presented the results of a series of structured interviews carried out to analyse the validity of the methodologies presented in this thesis. From the results shown in Figure 44 on page 242 it was seen that there was evidence for the validity and utility of the methodologies as interpreted by the individuals interviewed. There was also evidence that the detailed sub-sections, (Decision Support Questions, Indicative Ratios, Top Level MPC Systems Documentation Template), were considered valid by the interviewees, though the perceived validity varied according to the systems development experience of each interviewee.

Whilst the interviews presented evidence for the validity of the methodology and the decision support tools, it was decided that it was important to apply the methodology in different environments. For this reason Chapter 4 also helped to validate the Systems Development Life Cycle Methodology by presenting an analysis of three applications of the methodology. The three companies worked with were from different industries but all used MRPII systems to manage their businesses. The methodology was applied to help resolve real business problems within the companies and as a result of applying the methodology different solutions were adopted by the companies. The resulting solutions have been successfully applied by each of the companies.

This thesis has shown that both of the research themes addressed by this thesis have satisfied genuine business requirements within the collaborating company, by providing a number of new systems supporting operation of the business. There are a number of other companies that have been investigated by the author that could also benefit from these systems and from the methodologies derived in this thesis. Section 5.1.3 will emphasise the key points arising from this research.

5.1.3 Key Elements of the Research

This work was industrially based and has developed a series of practical techniques and methodologies to assist with the structured development and documentation of Manufacturing Planning and Control systems. These methodologies were developed as a result of reflection upon development projects undertaken in real-life situations. The companies worked with were all manufacturing companies and all used Manufacturing Resource Planning software as their main Manufacturing Planning and Control system. The five key elements of the research were as below:

1. **Satellite Systems;**

   This work has shown that there are a number of techniques that could be used when developing Manufacturing Planning and Control systems. One of the options available to systems developers is to use a development of the hybrid approach to systems development, the 'satellite systems' approach. Two different types of 'satellite system' were identified; integrated satellites and joined satellites and it was seen that the definition of the system core is key to this approach. It was shown that the satellite systems approach has been successfully applied in a number of companies.

2. **The Systems Development Life Cycle Methodology;**

   A methodology was developed from experience gained whilst undertaking systems development projects within the collaborating company. The methodology guides systems developers when they are choosing which option to apply when undertaking an MPC systems development project. Some decision support tools were also developed to assist with the application of the methodology. It was shown that the methodology has been successfully applied in a number of companies.

3. **The Four Levels of Documentation;**

   When undertaking the systems development projects within the collaborating company it was concluded that systems documentation needs to be created to complement any MPC systems development activity. It was seen that effective documentation requires a multi-level approach to the document creation process and that there are four levels at which an MPC system should be documented.

4. **The Top Level MPC System Documentation Template;**

   Whilst the four levels of documentation help the documentation author to determine how to document their MPC systems it is more difficult to determine what should be documented. A top level MPC system documentation template was created to help identify the documentation topics that should be considered. This template was derived from exposure within a discrete batch manufacturing environment and it was emphasised that the template should be modified to fit the business in question when producing documentation.

5. **The New Systems Development Life Cycle Methodology.**

   Recognising the direct links between systems development and documentation the systems development life cycle methodology was expanded to include documentation. This extension of the methodology included the documentation tools described above and also included feedback between the documentation process and the systems development process.

Section 5.2 will present the conclusions that can be drawn from this thesis and will discuss whether the aims and objectives of the Total Technology PhD project were met.

5.2 Conclusions

An analysis of the work presented in this thesis yields the following conclusions:

An MPC system core should not be modified unless absolutely necessary;

It has been shown that there is likely to be a significant threat to system integrity and complexity when undertaking modifications to an MPC system core. This can lead to unstable systems and to problems when trying to upgrade the core.

There are satisfactory alternatives to avoid having to modify an MPC system core;

A number of different approaches to the development of MPC systems have been identified. The satellite systems approach was developed to help systems developers when changes needed to be made to an MPC system that might otherwise have required a modification to the core. The satellite systems approach was successfully applied in the collaborating company and by a number of case study companies.

Before developing MPC systems determine whether the process can be changed;

Manufacturing Planning and Control systems are invariably changed in order to satisfy business process requirements. In some instances it is possible to modify the business process and in doing so to avoid having to develop the existing systems.

It is important that systems development activity is structured;

It has been shown from interview that systems practitioners believe that there is a need for a structuring of the systems development process. This structured approach is necessary to help systems developers ensure that the most appropriate solution is adopted when considering MPC system development projects.

A methodology for MPC systems development projects can be developed;

In order to provide a structured approach to MPC systems development a systems development methodology has been developed from observation whilst undertaking systems development activity within a collaborating company. This methodology has been successfully applied in a number of case study companies.

The systems development methodology needs decision support tools;

The systems development methodology requires a systems development team to select the appropriate option from a structured sequence of options. It was recognised that the methodology on its own was not enough and needed some decision support tools to simplify the choice of systems development technique. The Decision Support Questions and the Indicative Ratios were developed for this purpose.

Documentation is an integral part of the MPC systems development process;

It has been shown that MPC systems development drives the need for documentation and that creation of this systems documentation can feed back to prompt further systems development. Following a detailed investigation it was shown that when documenting MPC systems four levels of documentation are necessary; documents that describe the purpose of a system, documents that describe the interactions within a business, documents that describe the detailed procedures and documents that describe the detailed operation of the software.

The systems development methodology can be extended to include documentation.

A systems documentation project was undertaken at the collaborating company and a structured approach to documentation was developed. This approach consisted of an extension of the systems development methodology to include documentation.

The original hypothesis was that it is possible to define a methodology for the structured development of manufacturing planning and control systems. This thesis has demonstrated that it is indeed possible to define such a methodology. Furthermore the thesis has shown that the systems development methodology needs to be extended to include the structured documentation of the systems and has presented an enhanced methodology that incorporates both systems development and documentation.

A number of objectives were developed to help the researcher structure the investigation of the hypothesis. From the summary of the work presented in section 5.1 and the detailed conclusions presented above it can be concluded that the original aims and objectives of the project were satisfied, this is outlined below:

- **Gain an understanding of MPC systems, by literature review and case-study;**

  The literature review was presented in Chapter 2 and the case studies in section 3.1.
- **Gain an understanding of the Ferodo MPC systems;**

  The systems were investigated and were explained in sections 1.3 and 3.3.
- **Develop the Ferodo MPC systems to increase their effectivity;**

  The systems were developed and the new systems were presented in section 3.4.
- **Devise a methodology for MPC systems development;**

  The methodology was developed and presented in sections 3.5.2 and 3.11.3.
- **Test the methodology in order to help verify that the methodology has utility.**

  Structured interviews and methodology applications were presented in Chapter 4.

In addition to satisfying the research objectives the project satisfied the extra objectives of the Total Technology scheme as the collaborating company benefited from the output of the research project and the researcher gained significant practical experience.

5.3 Recommendations for Further Work

There have been two themes discussed in this thesis:

1. Structured Development of MPC Systems;
2. Structured Documentation of MPC Systems.

Both of these themes could be taken further as below.

5.3.1 The Structured Development of MPC Systems

Further Validation

The Systems Development Life Cycle Methodology and the decision support tools were derived from an understanding of MPC systems. They were validated in three ways:

1. Analytical Discussion;
2. Structured Interview;
3. Three applications of the System Development Life Cycle Methodology.

Both the Systems Development Life Cycle Methodology and the decision support tools could be further validated by further application in different scenarios.

Indicative Ratios

The Indicative Ratios rely on the prior experience of the user to help determine the relative ratio between the transaction rates. The Indicative Ratios could be applied in a number of different scenarios and different companies to help derive absolute numerical values for the Indicative Ratios, allowing use by less experienced users. The Indicative Ratios numerical values may vary by company type and therefore an initial selection process may need to be built into the Indicative Ratios decision support tool.

5.3.2 The Structured Documentation of MPC Systems

Further Validation

As with the Structured Development of MPC Systems the expansion of the Systems Development Life Cycle Methodology could be further validated by application in a number of different scenarios and different companies.

The Top Level Systems Documentation Template

The New Systems Development Life Cycle Methodology relies on the customisation of the Top Level MPC System Documentation Template to match the systems being documented. This process would be simplified if a number of more detailed templates could be developed for specific types of business, to allow the user to 'Fill in the Blanks'.

The Documentation Publishing Medium

Currently MRPII databases contain software documentation in the form of help text which is freely accessible. It would be of benefit if this could be extended so that the users can store and access their own documentation within the MRPII system.

This could be achieved if each MRPII screen allowed the entry of user configurable text. This in itself would not contribute to knowledge. Once this is done, however, a technique for the verification of the data could be developed, possibly using a pre-defined structure for the documentation and check boxes, as there are a finite number of configurations of the options available in an MRPII system. With these check boxes the system could verify that the implications of an option selected in one part of the system are reflected elsewhere.

Chapter 6. Appendices

6.1 Quality Analysis Files

6.1.1 Main Datafile

This datafile uses the Work Order Identification Number, Work Order ID, as its key or index field. This field allows easy identification of the Work Order that commissioned the work from the factory and thereby provides details of the items in question, date and quantity ordered. This number acts as the basis of the operation of the manufacturing modules of the core system as each batch of material that is processed in the factory gets its own unique Work Order ID field. From the Work Order ID, the system can look up the product line of the items being made. This is a secondary key field that allows a link with the control file. With these fields the following information can stored for each batch of product in the factory:

- Work Order ID Number (*key field*);
- Date of modification;
- Product line (*secondary key field*);
- Number of INCs of type 1;
- Number of INCs of type 2;
- Number of INCs of type 20;
- Associated text 1;
- Associated text 2;
- Associated text 3.

<u>6.1.2 Headings Control File</u>

This type of file is called a control file as the data stored in it is static, (it does not change frequently) and controls the display of data to allow users to understand the data as meaningful information. The headings that should be displayed against each INC type and the associated text fields are stored in this control file, split up by product line, to allow different INC types to be specified for different product types. In addition to the stored headings, this control file contains a logical field that allows the users to specify individually whether each INC type should be included in the calculations that were explained in section 3.4.1. The fields stored in this control file, indexed by product line, are:

- Product line (*Key Field*);
- Heading for INCs of type 1;
- INCs of type 1 to be included in calculations?, (*y/n stored*);
- Heading for INCs of type 2;
- INCs of type 2 to be included in calculations?, (*y/n stored*);
- Heading for INCs of type 3;
- INCs of type 3 to be included in calculations?, (*y/n stored*);
- Heading for INCs of type 20;
- INCs of type 20 to be included in calculations?, (*y/n stored*);
- Heading for associated text 1;
- Heading for associated text 2;
- Heading for associated text 3.

6.2 Statistical Process Control, (SPC)

One of the tools used by SPC systems is the definition of control limits. Any item that is being tested against a specification has limits outside of which is it considered to be defective. These limits are the specification limits. SPC rules are measured against control limits, which themselves are set within the specification limits, (this explanation will consider the system in question to be computational, the process can be done manually). There are two types of control limit to calculate:

1. The average mean upper and lower control limits, (UCL_{Mean} and LCL_{Mean});
2. The average range upper and lower control limits, (UCL_{Range} and LCL_{Range}).

Both of these are themselves subdivided into two types:

1. Tests with multiple results taken;
2. Tests with one result taken.

The SPC system needs to be told how many immediately past means or ranges are required to calculate the averages used to calculate the control limits, (n) and how many test results to take, (m). Once it is determined whether multiple or individual values are to be taken the means and ranges need to be calculated. With these means and ranges the control limits can be calculated and the mean and range in question analysed against them, (if the mean and ranges fall outside of the control limits the first test has failed).

The means, ranges and control limits are calculated as follows.

In the case of **multiple** results taken:

x = The mean, which is the test mean, (*the test result*);

r = The range, which is the test max - test min;

X = The average mean, which is the sum of all the required means divided by the number of required means;

R = The average range, which is the sum of all the required ranges divided by the number of required ranges.

Therefore, in the case of **m** results being taken and **n** past means required.

$$x = \frac{val_1 + val_2 + \ldots\ldots + val_m}{m} \qquad r = val_{max} - val_{min}$$

$$X = \frac{x_1 + x_2 + \ldots\ldots + x_n}{n} \qquad R = \frac{r_1 + r_2 + \ldots\ldots r_n}{n}$$

In the case of **individual** results taken;

x = The mean, which is the test mean, (*the test result*);

r = The range, the test mean for the current test - the test mean for the previous test;

X = The average mean, which is the sum of all the required means divided by the number of required means;

R = The average range, which is the sum of all the required ranges divided by the number of required ranges.

The formulas applied in the case of **n** past means required, (**m**=1), are as follows. (Note that in the case of individual test results taken the range r must be converted to a positive value if it comes out negative).

$x = val$ $r = val_{current} - val_{previous}$

$$X = \frac{x_1 + x_2 + \ldots\ldots + x_n}{n}$$ $$R = \frac{r_1 + r_2 + \ldots\ldots r_n}{n}$$

With these values the four control limits can be calculated using the formulae and constants below.

Control Limit	Multiple Results	Individuals
UCL_{Mean}	$X + A_2R$	$X + E_2R$
LCL_{Mean}	$X - A_2R$	$X - E_2R$
UCL_{Range}	D_4R	
LCL_{Range}	D_3R	

A2, E2, D3 and D4 can be found from the table of constants below, reading across from the value of n found as above. It can be seen that n must be greater than 1 and that if n is less than or equal to 6 then the lower range control limit will not exist.

No used for control limits (n)	A_2	E_2	D_3	D_4
2	1.880	2.660	*	3.267
3	1.023	1.772	*	2.574
4	0.729	1.457	*	2.282
5	0.577	1.290	*	2.114
6	0.483	1.184	*	2.004
7	0.419	1.109	0.076	1.924
8	0.373	1.054	0.136	1.864
9	0.337	1.010	0.184	1.816
10	0.308	0.975	0.223	1.777
11	0.285		0.256	1.744
12	0.266		0.283	1.717
13	0.249		0.307	1.693
14	0.235		0.328	1.672
15	0.223		0.347	1.653
16	0.212		0.363	1.637
17	0.203		0.378	1.622
18	0.194		0.391	1.608
19	0.187		0.403	1.597
20	0.180		0.415	1.585
21	0.173		0.425	1.575
22	0.167		0.434	1.566
23	0.162		0.443	1.557
24	0.157		0.451	1.548
25	0.153		0.459	1.541

6.3 Property Recording Files

6.3.1 SPC Rules Control File

On implementation of the Property Recording system, the first file that needs data to be entered is the SPC Rules Control File, (if SPC is to be performed). This control file allows definition of one thousand SPC trend analysis scenarios with an SPC reference number as a key field. This is in contrast to the original systems used within Ferodo which only used one trend analysis scenario and so the new systems offered the benefits of detailed tuning of SPC requirements.

It was decided that there will be situations where SPC analyses will not be required. For this reason the SPC value of '000' was set aside to indicate that SPC should not be run when this code was discovered.

The SPC control file does not differentiate between ranges and averages. If, in the item tests control file, the user specifies that the system is to test SPC for both averages and ranges then the system will test the same SPC trend analysis scenario against both of them.

The following SPC rules can be defined and stored in the SPC rules control file:

- SPC reference number (*key field*);
- The number of consecutive historical readings not above 2/3 of upper control limit;

- The number of consecutive historical readings not below 2/3 of upper control limit;
- The number of consecutive historical readings not above 1/3 of lower control limit;
- The number of consecutive historical readings not below 1/3 of lower control limit;
- The number of consecutive historical readings not to be above the mean value;
- The number of consecutive historical readings not to be below the mean value;
- The number of historical readings not to be within 1/3 of the mean value;
- The number of historical readings not to be within 1/3 of the mean value;
- The number of consecutive historical readings not to be increasing in value;
- The number of consecutive historical readings not to be decreasing in value;
- The number of consecutive historical readings not to be alternating about the average;
- The number of historical readings to use when running analyses, (m);
- The number of historical readings to calculate control limits and average value, (n).

6.3.2 Standard Tests Control File

The Standard Tests Control File should be the second file configured. This file, using a Standard Test Number as a key field allows the users to specify the number of test readings to take for each type of test that is performed in the factory and to link the Standard Tests to the SPC trends analysis scenario via the SPC reference number. Whether a control limit should be based upon multiple results, (m>1), or individual, (m=1), can be determined from the 'number of readings to take' field in this control file. A value of 1 indicates individual reading, any other value indicates multiple readings, (m).

The structure of this control file is as follows:

- Standard Test number (*key field*);
- Short description;
- Test description 1 - 4;
- Number of results to average, (*up to 20*);
- Text to accompany four spare numerical fields, (*1-4*);
- Text to accompany four spare descriptive fields, (*1-4*);
- SPC reference number;
- Unit of measure.

6.3.3 Item Tests Control File

The Item Tests Control File should be the third and final file configured. The control file allows for up to fifty Standard Tests to be associated with each item that is dealt with by the factory, both purchased and manufactured. The system will not allow items to be defined if they do not exist in the core system. In addition to allowing the definition of tests that should be carried out, the control file accepts specification limits and a nominal value as a guide for the user. The control file also allows the user to specify whether SPC should be performed on ranges, averages, both or neither. Finally the control file allows the user to indicate whether each test is compulsory or to be done by statistical sampling.

The structure of this control file is as follows:

- Item Number (*key field*);
- Ranges, averages, both or neither, (*r/a/b/n stored*);
- Compulsory test for test 1?, (*y/n stored*);
- Standard test number for test 1;
- Upper specification limit for test 1;
- Lower specification limit for test 1;
- Nominal value for test 1;

- Compulsory test for test 50?, (*y/n stored*);
- Standard test number for test 50;
- Upper specification limit for test 50;
- Lower specification limit for test 50;
- Nominal value for test 50.

6.3.4 Main Data Storage Datafile

It is the Main Data Storage Datafile that brings all the control file information together. One of the main design decisions made was that this datafile was to copy important information from the control files when data was being stored in order to accurately put together history in the event of future control file content changes.

The structure is as follows:

- Data type, (*manufactured or purchased*);
- Primary identifier, (*Work Order ID or Transaction History Number - key field*);
- Standard test number;
- Item Number;
- Test results, (*up to 20 stored*);
- Test mean, maximum and minimum values;
- Test range;
- Four spare numerical fields, (*1-4*);
- Four spare descriptive fields, (*1-4*);
- A large spare discussion field;
- Date;
- Time;
- Upper control limit - mean;
- Lower control limit - mean;
- Upper control limit - range;
- Lower control limit - range;
- Was a retest necessary, (*y/n stored*);
- Was the item failed due to violation of absolute specification limits, (*y/n stored*);
- User ID of the person keying in the data.

6.3.5 Data Entry and Reporting

The system has a number of data entry screens and reports available:

- Outstanding Tasks Report;
- Completed Tasks Report;
- SPC Trend Analysis Warnings Report;
- SPC Trend Analysis Warning Action Entry Screen;
- Item Failure Entry Screen;
- Item Test Results Data Entry Screen;
- Data Download For Spreadsheet Analysis Screen.

This section will look at each of these in turn.

Outstanding Tasks Report

This report starts by asking the user to specify whether the report should look at purchased or manufactured items. It then asks the user to specify what sort of outstanding tasks to look at. This is performed using a report generator screen. For example for purchased items the following report generator selections appear:

Item Number	From:	To:
Effective Date of receipt into stores	From: / /	To: / /
Product Line	From:	To:
Site	From:	To:
Standard Test Number	From:	To: 999
Receiving location	From:	To:
Transaction History Number	From:	To: 99999999

Based on the criteria input the system interrogates the core system and determines the list of Purchase Orders that match the criteria input. The system then looks at the main datafile to see if there are any Purchase Orders that have data entered against all the required tests and if there are any that have failed specification limits. These Purchase Orders are removed from the list. The list is then presented to the user either on-screen or printed. A similar process is undertaken for the Work Orders.

Completed Tasks Report

The completed tasks report works in a similar way to the outstanding tasks report with a similar report generator but it only displays those tests with data entered against all the required tests, (any items with tests that have failed specification limits are not included).

Note that there is no report to show the items that have failed specification limits as these are actually INCs in the case of manufactured items or returned goods in the case of purchased items. The purpose of entering this information into this system is simply to remove the items from the outstanding tasks report.

SPC Trend Analysis Warnings Report

Whenever a test fails the SPC rules it is being tested against a warning gets put into the SPC Trend Analysis Warning file. This report displays the warnings received that have not been dealt with.

SPC Trend Analysis Warning Action Entry Screen

This screen allows a user to indicate that a warning has been dealt with.

Item Failure Entry Screen

This screen allows a user to log the fact that an item has persistently failed specification limits and did not pass on retest. The system will mark all records stored against the item for the Work Order or Purchase Order in question and will activate the item failure flag in the main control file. An item indicated in this way will not appear in the outstanding or completed tests report and it will not be used for SPC analyses either.

Item Test Results Data Entry Screen

Either by using the Work Order paperwork or the Outstanding Tasks report the necessary identification and key field data can be entered into the main data entry screen to identify an item and a batch of product - purchased or manufactured. The system requires more than one identifier to be entered to ensure that the user is entering the correct test results by cross-referencing with the core system.

From the data input in this way the system will determine the standard tests required, the specification data and if each test is required. The following input screen is displayed:

Item Number________ Order Number_________

Property Description	Value	Req (y/n)	Min	Max	Unit of Measure
Test 1		y	a	b	mm
Test 2		n	c	d	g/cc

The system actually displays up to 50 tests, 16 at a time, on screen. In the display given above there are two tests, test 1 is required, is measured in millimetres and has a minimum specification value of ‘a’ and a maximum value of ‘b’. Test 2 is not required for every batch, is measured in grams per cubic centimetre, has a minimum specification value of ‘c’ and a maximum value of ‘d’.

The user will press the tab key to select the appropriate test and then by pressing enter will open the data entry screen as a window, on top of the main screen, leaving the line containing the Item Number exposed. This screen will allow the entry of individual data elements. The appearance of this window will be created by the system based on the rules contained in the database. The rules in the database may not require the entry of 20 tests and only the number required will be displayed, as below.

Short Description______		Standard Test Number_____	
Test Description 1_______		Spare Numerical Text 1__________	
Test Description 2_______		Spare Numerical Text 2__________	
Test Description 3_______		Spare Numerical Text 3__________	
Test Description 4_______		Spare Numerical Text 4__________	
Test Result 1_____	Test Result 4_____	Test Result 7_____	Test Result 10_____
Test Result 2_____	Test Result 5_____	Test Result 8_____	Test Result 11_____
Test Result 3_____	Test Result 6_____	Test Result 9_____	Test Result 12_____
Spare Descriptive Text 1__________		Spare Descriptive Text 2__________	
Spare Descriptive Text 3__________		Spare Descriptive Text 4__________	

Discussion____

Calculated Mean_____ Re-Test (Y/N)____

When the results have been entered pressing return will cause the system to calculate an average and if all the results are within the absolute limits the cursor will move on to the retests y/n box. If any results are outside of the absolute limits the system will prompt the user to re-enter the results that fail the absolute limits as a retest.

If a retest was performed the retests yes/no box will be filled in as a yes, if not the retests yes/no box will be left as no, this is for information purposes only. On exiting the window, the cursor will move to the next field of the main input screen and data will be saved at this point. The mean value of the test results will be displayed in the val. column of the first screen.

Also at this point the system will look at whether ranges, averages, both or neither are to be tested for SPC limits and will perform SPC based on the trend analysis scenarios configured. If the system has been instructed to work with 10 historical records and only 8 exist the system will run SPC with the 8 records only, (the system will not run SPC at all if there are 6 or fewer historical records). Any violations of the trend analysis will be displayed on screen and will be written into the rules violation file.

Data Download For Spreadsheet Analysis Screen

This screen uses a report generator similar to the outstanding tasks report to allow extraction of some or all of the main datafile to a spreadsheet file along with descriptive fields from the control files. Once in the spreadsheet the data can be manipulated and sent on to the original Quality Analyst SPC package if more detailed analyses and graphical displays are required.

6.4 Manufacturing Instructions Files

6.4.1 Index Headings Control File

As a result of designing the Manufacturing Instructions systems as a separate sub-system located within the core it was possible to allow for total user definition of the database structure. The database was designed to allow the users to define their own index to uniquely identify item characteristics.

Ferodo, for example, use physically separate MFG/PRO databases in each of the divisions explained in section 1.3. If this was not the case then all the divisions would be controlled using one database and logically separated inside the database. In these two cases the Manufacturing Instructions would have different indexes as in the second case the index would have to define the division in question first. An additional advantage of this approach is that international users can change the index headings to reflect their own language, (even when the database is full of data).

The index headings control file that allows this has the following structure:

- Index heading descriptions, (*1 - 10*) (*10 key fields*);
- Template label to allow definition of the word used to describe a template;
- Change label to allow definition of the word used to describe the revision number.

Once these headings have been configured it is then possible to define a number of templates of fields that can be stored against the indexes. For example it is possible to define any item manufactured in the Ferodo OE division with the following headings:

- Product line;
- Mix number;
- Die / tooling reference;
- Item Number;
- Customer reference for the item, (*used when initially configuring an item*).

There are, however, a number of different product lines including:

- Commercial Vehicles;
- Press Formed Car Disc Brake Pads;
- Press Cured Car Disc Brake Pads;
- Drum Brake Linings.

The same index structure will uniquely identify any item from any of the product lines. Each product line, however, has different data fields as the processes are physically separate. Using a template attached to the index it is possible to identify the data field headings that should be used with the data, (to convert it from data into information).

6.4.2 Data Field Headings Control File

The data field headings control file is uniquely identified by a template number.

There are 100 possible template numbers, each of which can store 100 numerical fields and 100 text fields. The structure is as follows:

- Template number (*key field*);
- Numerical field heading, (*1 - 100*);
- Text field heading, (*1 - 100*).

6.4.3 Main Data Storage Datafile

Once the Index, Data fields and Change and Template headings have been defined the main data storage datafile allows data to be stored and associated with these headings. The structure is as follows:

- Heading values, (*1 - 10*) (*10 key fields*);
- Template number;
- Date;
- Time;
- Change or revision number;
- Numerical field value, (*1 - 100*);
- Text field value, (*1 - 100*).

Figure 50 below better illustrates the relationship between the different databases. The figure shows the various headings across the top of a simple database, similar in appearance to a spreadsheet with headings.

Figure 50: Manufacturing Instructions Main Datafile Structure

Index Headings							T	C	Data Field Headings						
1	2	3	4	5	⇒	10	1	1	1	2	3	4	5	⇒	100

Key:

T = Template number;

C = Change / revision number.

6.4.4 History Datafile

In addition to the main datafiles a history datafile was required in the original definition of requirements. The history file was designed to store an entire datafile row in the event of a change to the information contained. In addition to this the field headings are stored to cater for changes to the datafile headings and templates.

It has already been mentioned that change control over the Manufacturing Instructions system was to be done using the existing Change Control Form. It is vital that this number is stored against any change made to identify the origins of the change. Rather than making this a systems issue and designing it into the operation of the database it was decided that this should be an issue for the users of the system. Therefore, in the OE implementation the first data field is used to store the current Change Control Form number and each time a change is made the Change Control Form number is updated in the datafile.

The history datafile structure is as follows:

- Heading value, (*1 - 10*) (*10 key fields*);
- Change value;
- Template number;
- Date;
- Time;
- Index headings, (*1 - 10*);
- Template heading;
- Change heading;
- Numerical field heading, (*1 - 100*);
- Numerical field value, (*1 - 100*);
- Text field heading, (*1 - 100*);
- Text field value, (*1 - 100*).

6.4.5 Reporting

When a user wishes to read information from the Manufacturing Instructions database they can use the report generator screen or download the data to a spreadsheet for analysis.

The input screen looks like below:

Main Datafile or History File Required, (M/H):

Data Heading 1 From: To:

Data Heading 10 From: To:

Template Heading From: To:

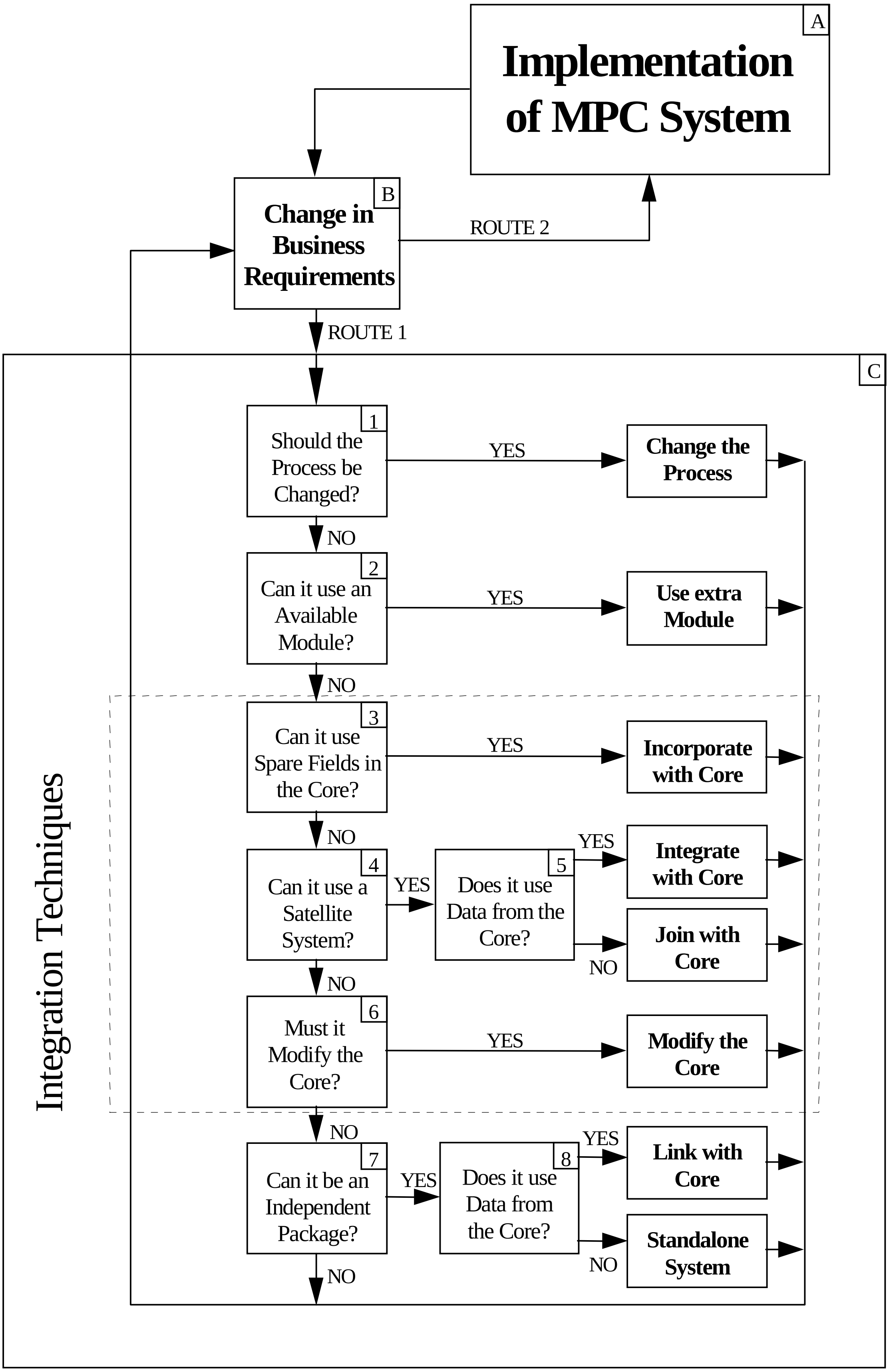
Change Heading From: To:

Date From: / / To: / /

The output appears, either displayed on screen or printed and contains all the information contained in the database about the item or range of items selected. There is an option to restore historical data if required.

6.5 The Structured Interview Template

The Systems Development Life Cycle Methodology



Systems Development Life Cycle Process Step Descriptions

ROUTE 1:	**(Preferred Route - Systems Development)**
Change the Process	Change the process to fit in with existing systems
Use Extra Module	Purchase a new module from the core system suppliers
Incorporate with Core	Incorporate new data using spare fields within the core
Integrate with Core	Integrate a satellite system with the core, (with data lookup)
Join with Core	Join a satellite system with the core, (with no data lookup)
Modify the Core	Modify the core to perform a new, required function
Link with Core	Link an independent package with the core
Standalone System	Create a standalone system with an independent package
ROUTE 2	**(Done rarely, to use new technology, or if Route 1 fails)**
Implementation	Implement a new core using existing techniques

Decision Support Questions

Should the Process be Changed?

1 Is it possible to change the process to meet the changed business requirements, (taking into account external accreditation and quality guidelines)?
2 If so, can the new process be used without causing problems elsewhere in the systems that cannot be easily addressed by this methodology?

Can it use an Available Module?

1 Is there a module available?
2 Does the module contain sufficient functionality?
3 Is the module affordable?
4 Are the processes within the module simple enough to use, (such as not causing too many transactions, requiring additional staff to be taken on or overburdening existing staff)?

Can it use Spare Fields in the Core?

1. Are there any applicable spare fields in the core, (that store data in the same format, preferably using the same indexes)?
2. If so, are there enough of these spare fields to use for the new system?

Can it use a Satellite System?

1 Can an auxiliary database be written to meet business requirements, (remembering the link into the existing systems)?
2 Is there sufficient programming resource available to do the work?
3 Can the work be done within budget?

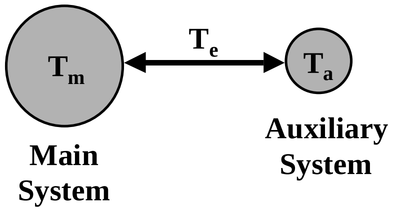
Must it Modify the Core?

1 Can modifications be made to the core to meet the business needs?
2 Is there sufficient programming resource available to do the work?
3 Can the work be done within budget, (remembering the continuing cost of each upgrade)?

Can it be an Independent Package?

1 Is there an independent package available?
2 Does the independent package contain sufficient functionality?
3 Is the independent package affordable?
4 In the event of a linked system can the independent package link to the core?
5 Can this be achieved at an affordable cost?

Decision Support Indicative Ratios

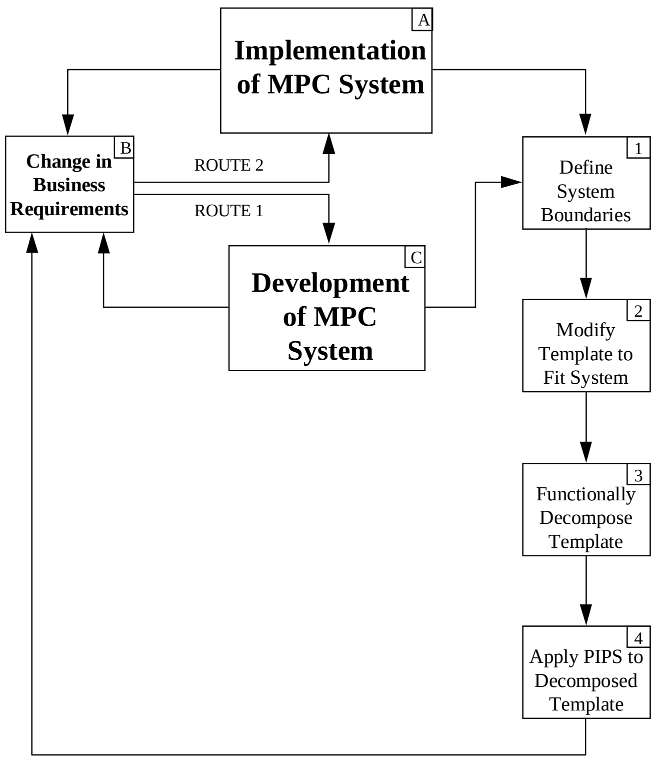


Definitions
T_m = Transactions rate in main system
T_a = Transaction rate in auxiliary system
T_e = Transaction rate between systems

1. If $T_m/T_a = \infty$ then no need for new system, (as $T_a = 0$, i.e. No Auxiliary System);
2. If $T_m/T_a >> 1$ then Incorporated System;
3. If $T_m/T_a > 1$ then:
   3.1. If $T_a/T_e = \infty$ then Separate System / Joined Satellite System, (as $T_e = 0$);
   3.2. If $T_a/T_e >> 1$ then Separate System / Joined Satellite System;
   3.3. If $T_a/T_e > 1$ then Integrated Satellite System;
   3.4. If $T_a/T_e = 1$ then Integrated Satellite System;
   3.5. If $T_a/T_e < 1$ then Integrated Satellite System;
   3.6. If $T_a/T_e << 1$ then Modified Core.
   3.7. If $T_a/T_e = 0$ then Modified Core, (as $T_e = \infty$).
4. If $T_m/T_a = 1$ then New Core;
5. If $T_m/T_a < 1$ then New Core;
6. If $T_m/T_a << 1$ then New Core;
7. If $T_m/T_a = 0$ then Implement Core for first time, (as $T_m = 0$, i.e. No Core System).

Assumption: $T_a \propto T_e$

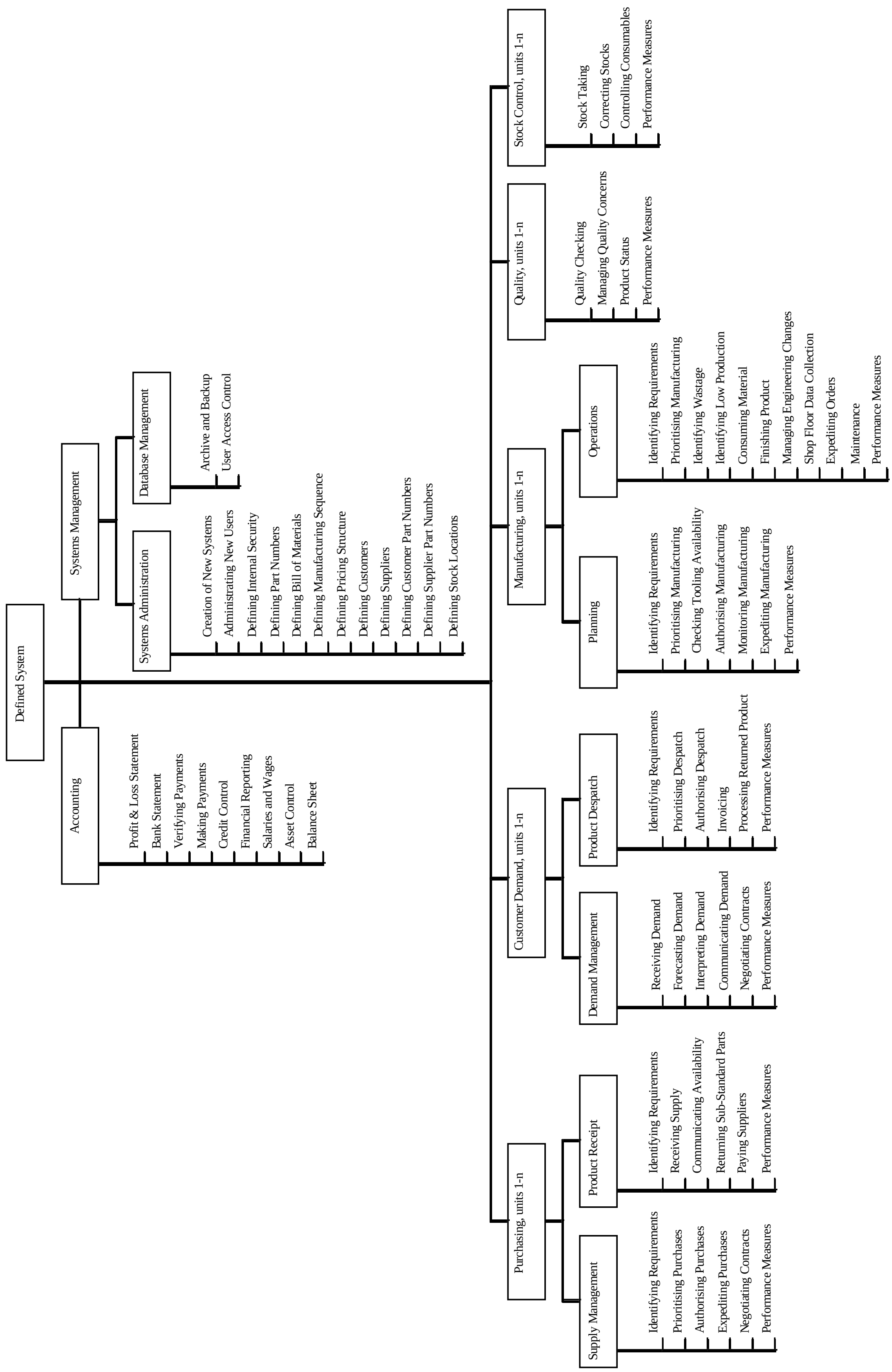
The Systems Development Life Cycle with Documentation



PIPS

1. Documents that describe the **Purpose** of the system;
2. Documents that describe the **Interaction** within and between departments;
3. Documents that describe the **Procedures** followed within departments;
4. Documents that describe the **Software** used to control the business.

Top Level MPC Systems Documentation Template



Interview Agenda

1. Determine job description

Purpose: General Information
Output: Categorisation

2. Determine systems analysis experience

Purpose: General Information
Output: Categorisation

3. Determine if they have experience of MRPII systems

Purpose: General Information
Output: Categorisation

4. Explain MRPII systems if required
   Purpose: To enable discussion of the Systems Development Life Cycle Methodology as applied to MRPII systems
   Output: None

5. Define the definition of Methodology as applied in this work
   Purpose: To enable discussion of the Systems Development Life Cycle Methodology as applied to MRPII systems
   Output: None

6. Determine if they believe a formal Methodology would help systems development activities
   6.1. In your opinion would a formal Methodology assist with Systems Development Activities? (**<u>Yes</u>**/**<u>No</u>**)
   6.2. If not why not?

   Purpose: To validate the requirement for a formal systems development methodology
   Output: Requirement or No-Requirement

7. Present the Systems Development Life Cycle Methodology, the Decision Support Questions, Indicative Ratios and the Systems Development Life Cycle with Documentation

Purpose: To enable discussion of the Systems Development Life Cycle Methodology as applied to MRPII systems
Output: None

8. Discuss each step in methodology and the relative positions between each step and the next / previous ones

8.1. In your opinion is changing the process preferable to modifying the core system? (**<u>Yes</u>**/**<u>No</u>**)
8.2. If not why not?

____

____

8.3. In your opinion is the use of satellite systems preferable to modifying the core system? (**<u>Yes</u>**/**<u>No</u>**)
8.4. If not why not?

____

____

8.5. In your opinion is the sequence of steps as a whole appropriate and complete? (**<u>Yes</u>**/**<u>No</u>**)
8.6. If not why not?

____

____

8.7. In your opinion is the Systems Development Life Cycle Methodology **<u>Validated</u>**, Should it be **<u>Modified</u>** or do you **<u>Reject</u>** the Methodology

Purpose: To validate the construction of the Systems Development Life Cycle Methodology
Output: Methodology Validated, Modified, Rejected

9. Determine whether the questions would help / provide assistance if they were: Experienced, intermediate, novice.

9.1. If you were inexperienced in systems analysis and design would the structured questions assist with applying the Systems Development Life Cycle Methodology? (**<u>Yes</u>**/**<u>No</u>**)
9.2. If not why not?

____

____

9.3. If you had limited experience in systems analysis and design would the structured questions assist with applying the Systems Development Life Cycle Methodology? (**<u>Yes</u>**/**<u>No</u>**)

9.4. If not why not?

____

____

9.5. If you were experienced in systems analysis and design would the structured questions assist with applying the Systems Development Life Cycle Methodology? (**<u>Yes</u>**/**<u>No</u>**)

9.6. If not why not?

____

____

9.7. In your opinion are the structured questions **<u>Validated</u>**, Should they be **<u>Modified</u>** or do you **<u>Reject</u>** them?

Purpose: To determine the applicability of the top level Decision Support Questions
Output: Questions Validated, Modified, Rejected. Understanding gained of experience level required

10. Determine whether the Indicative Ratios would help / provide assistance if they were: Experienced, intermediate, novice. Discuss the ratios with the experienced analysts.

10.1. If you were inexperienced in systems analysis and design would the Indicative Ratios assist with applying the Systems Development Life Cycle Methodology? (**<u>Yes</u>**/**<u>No</u>**)

10.2. If not why not?

____

____

10.3. If you had limited experience in systems analysis and design would the Indicative Ratios assist with applying the Systems Development Life Cycle Methodology? (**<u>Yes</u>**/**<u>No</u>**)

10.4. If not why not?

____

____

10.5. If you were experienced in systems analysis and design would the Indicative Ratios assist with applying the Systems Development Life Cycle Methodology? (**<u>Yes</u>**/**<u>No</u>**)

10.6. If not why not?

____

____

10.7. (If experienced in systems analysis) In your opinion are the ratios realistic? (**Yes**/**No**)

10.8. In your opinion are the structured questions **Validated**, Should they be **Modified** or do you **Reject** them?

Purpose: To determine the applicability of the Indicative Ratios
Output: Indicative Ratios Validated, Modified, Rejected. Understanding gained of experience level required

11. Discuss the extension of the Methodology to include documentation

11.1. In your opinion should documentation be part of a Systems Development exercise? (**Yes**/**No**)

11.2. If not why not?

____

____

11.3. In your opinion is the sequence of steps as a whole appropriate and complete? (**Yes**/**No**)

11.4. If not why not?

____

____

11.5. In your opinion is the documentation extension to the Systems Development Life Cycle Methodology **Validated**, Should they be **Modified** or do you **Reject** them?

Purpose: To validate the construction of the Systems Development Life Cycle Methodology with Documentation
Output: Documentation addition Validated, Modified, Rejected

12. Determine whether the top level MPC systems documentation template would help / provide assistance if they were: Experienced, intermediate, novice.

12.1. If you were inexperienced in systems analysis and design would the top level MPC template assist you with documenting the Systems Development Life Cycle Methodology? (**Yes**/**No**)

12.2. If not why not?

____

____

12.3. If you had limited experienced in systems analysis and design would top level MPC template assist you with documenting the Systems Development Life Cycle Methodology? (**Yes**/**No**)

12.4. If not why not?

____

____

12.5. If you were experienced in systems analysis and design would the top level MPC template assist you with documenting the Systems Development Life Cycle Methodology? (**Yes**/**No**)

12.6. If not why not?

____

____

12.7. In your opinion is the top level MPC template **Validated**, Should it be **Modified** or do you **Reject** it?

Purpose: To determine the applicability of the MPC systems documentation template
Output: Top Level MPC systems documentation template Validated, Modified, Rejected

13. Determine whether the PIPS levels of documentation would help / provide assistance if they were: Experienced, intermediate, novice.

13.1. If you were inexperienced in systems analysis and design would the PIPS levels of documentation assist you with documenting the Systems Development Life Cycle Methodology? (**Yes**/**No**)

13.2. If not why not?

____

____

13.3. If you had limited experienced in systems analysis and design would the PIPS levels of documentation assist you with documenting the Systems Development Life Cycle Methodology? (**Yes**/**No**)

13.4. If not why not?

____

____

13.5. If you were experienced in systems analysis and design would the PIPS levels of documentation assist you with documenting the Systems Development Life Cycle Methodology? (**Yes**/**No**)

13.6. If not why not?

____

____

13.7. In your opinion are the PIPS levels of documentation **Validated**, Should they be **Modified** or do you **Reject** them?

Purpose: To determine the applicability of the PIPS levels of documentation
Output: PIPS levels of documentation Validated, Modified, Rejected

References

Agosti, M., Crestani, F. and Melucci, M. (1996), "Design and implementation of a tool for the automatic construction of hypertexts for information retrieval", Information Processing and Management, Vol. 32, No. 4, Pages 459-476.

Ang, J.S.K., Sum, C.C. and Yang, K.K. (1994), "MRP II company profile and implementation problems: A Singapore experience", International Journal of Production Economics, Vol. 34, No. 1, Pages 35-45.

Anon. (1995), "MRPs effectiveness is questioned", Hydrocarbon Processing, Vol. 74, No. 5, Page 33.

Avison, D.E. and Wood-Harper, A.T. (1990), "Multiview: An Exploration in Information Systems Development", Blackwell, Oxford, UK.

Baines, R. W, Small, C., Colquhoun, G. J. (1996), "Modelling methods in the manufacturing change process", Proceedings of the 12th NMCR Conference, The University of Bath, September.

Barua, A. and Ravindran, S. (1996), "Reengineering information sharing behaviour in organisations", Journal of Information Technology", Vol. 11, No. 3, Pages 261-272.

Baskerville, R.L. and Wood-Harper, A.T. (1996), "A critical perspective on action research as a method for information systems research", The Journal of Information Technology, Vol. 11, No. 3, Pages 235-246.

Baynton, K. (1996), "Biting the Intranet Bullet", Telecommunications (International Edition), Vol. 30, No. 12, Pages 39-43.

Berners-Lee, T., "Style Guide for online hypertext", CERN, http://www.w3.org/provider/style/.

Blood, B.E. (1992), “Your MRP System Is Basically Ineffective...and I Know Why!”, Annual International Conference Proceedings - American Production and Inventory Control Society, Pages 4-6.

Blum, F.H. (1955), “Action Research - a Scientific Approach?”, Philosophy of Science, Vol. 22, No. 1, Pages 1-7.

Bodington, C.E. (1995), “Planning, Scheduling, and Control Integration in the Process Industries”, McGraw Hill, New York, USA.

Borälv, E., Göransson, B., Olsson, E. and Sandblad, B. (1994), “Documentation and information services in the HELIOS project”, Computer Methods and Programs in Biomedicine, Vol. 45, Supplement, Pages S139-S145.

Borysowich, C.W. (1996), “Notes Takes on the Internet”, http://www.innergy.com/op-ed.htm.

Bradford, A.N. (1988), “A Planning Process for Online Information”, In: Doheny-Farina, S. (Ed), “Effective Documentation: What We Have Learned from Research”, MIT Press, Cambridge Massachusetts, USA, No. 9, Pages 185-211.

Brockmann, R.J. (1990), “Writing Better Computer User Documentation From Paper to Hypertext Version 2.0”, John Wiley and Sons Inc., New York, USA.

Browne, J., Harhen, J. and Shivnan, J. (1996), “Production Management Systems - An Integrated Perspective”, Addison Wesley, UK.

Bulkeley, D. (1996), “I Want All Our Documentation on Intranets”, DATAMATION, Vol. 42, No. 12, Page 56.

Burgoine, J. (1988), “Beyond MRPII”, Production Engineer, January, Pages 31-33.

Buyers, R. (1984), “Documentation of Business Procedures - A New Technique for an Old Problem”, Journal of Systems Management, Vol. 35, No. 10, Pages 16-19.

Carrie, A.S. and Banerjee, S.K. (1984), "Approaches to Implementing Manufacturing Information Systems", OMEGA The International Journal of Management Science, Vol. 12, No. 3, Pages 251-259.

Carroll, J.M., Smith-Kerker, P.L., Ford, J.R. and Mazur-Rimetz, S.A. (1988), "The Minimal Manual", In: Doheny-Farina, S. (Ed), "Effective Documentation: What We Have Learned from Research", MIT Press, Cambridge Massachusetts, USA, No. 4, Pages 73-102.

Cashmore, C. and Lyall, R. (1991), "Business Information: Systems and Strategies", Prentice Hall, London, UK.

Caswell, F. (1991), "Success in Statistics", 2nd Edition, John Murray (Publishers) Limited, London, UK.

Cerveny, R.P. and Scott, L.W. (1989), "A Survey of MRP Implementation", Production and Inventory Management Journal, 3rd Quarter, Pages 31-34.

Charney, D.H., Reder, L.M. and Wells, G.W. (1988), "Studies of Elaboration in Instructional Texts", In: Doheny-Farina, S. (Ed), "Effective Documentation: What We Have Learned from Research", MIT Press, Cambridge Massachusetts, USA, No. 3, Pages 47-72.

Checkland, P. (1981), "Systems Thinking Systems Practice", John Wiley and Sons, Chichester, UK.

Checkland, P. (1991), "From Framework through Experience to Learning: the essential nature of Action Research", In: Nissen, H., Klein, H.K. and Hirschheim, R. (Eds), Information Systems Research: Contemporary Approaches & Emergent Traditions, Elsevier Science Publishers, Amsterdam, The Netherlands.

Checkland, P. and Holwell, S. (1998), "Information, Systems and Information Systems - making sense of the field", John Wiley and Sons, Chichester UK.

Checkland, P. and Scholes, J. (1990), “Soft Systems Methodology in Action”, John Wiley and Sons, Chichester, UK.

Chrysler Corporation, Ford Motor Company and General Motors Corporation (1995a), “Quality System Requirements - QS-9000”, Second Edition, Carwin Continuous Ltd., Essex, UK.

Chrysler Corporation, Ford Motor Company and General Motors Corporation (1995b), “Statistical Process Control (SPC) Reference Manual”, Carwin Continuous Ltd., Essex, UK.

Clinton, B.D. and Hsu, K. (1997), “JIT and the Balanced Scorecard: Linking Manufacturing Control to Management Control”, Management Accounting, September, Pages 18-24.

Conway, M.G. (1993), “A Methodology for the Introduction of JIT Management to Support Functions”, PhD Thesis, UMIST.

Cox, D.R. (1990), “Quality and Reliability: Some Recent Developments and a Historical Perspective”, Journal of the Operations Research Society, Vol. 41, No. 2, Pages 95-101.

Dale, B.G. (1984), “Work Flow in Manufacturing Systems”, Engineering Management International, Vol. 3, Pages 3-13.

David, I. (1998), “Clocking in to the cyberfactory”, Professional Engineering, 16 September, Pages 33 and 35.

Dear, A. (1988), “Working Towards Just-in-Time”, Kogan Page Ltd., London, UK.

Debs, M.B. (1988), “A History of Advice: What Experts Have to Tell Us”, In: Doheny-Farina, S. (Ed), “Effective Documentation: What We Have Learned from Research”, MIT Press, Cambridge Massachusetts, USA, No. 1, Pages 11-24.

Denton, P.D. and Hodgson, A. (1997), "Systems Development for Global Manufacturing", Proceedings of the 5th International IEE Conference on Factory 2000, 2-4 April, Cambridge University, UK, Pages 395-402.

Desborough, J. (1996), "Report from the trenches", December, http://www.innergy.com/idm/w1n12/idm1296a.html.

Doniavi, A., Mileham, A.R. and Newnes, L.B. (1997), "Computer Integrated Systems Engineering in an Electronics Manufacturing Environment", Proceedings of the 13th NCMR Conference, Glasgow Caledonian University, 9-11 September, Pages 103-107.

Downs, E., Clare, P. and Coe, I. (1988), "Structured Systems Analysis and Design Method: Application and Context", Prentice Hall International Ltd., UK.

Dwyer, J. (1995), "Breathing life into MRP, Manufacturing Computer Solutions", July-August, Vol. 1, No. 6, Pages 37-41.

Edwards, J. and Gibson, P.R. (1997), "Integrated Multi-Media Computers in the Execution of ISO9000 Quality System Requirements for Document Control and Training", Computers & Industrial Engineering, Vol. 32, No. 3, Pages 529-538.

Eisenhartd, M.K. (1989), "Building Theories from Case Study Research", Academy of Management Review, Vol. 14, No. 4, Pages 532-550.

Eros, R. (1996), "Learning to Love the Intranet", Telecommunications (International Edition), Vol. 30, No. 6, Pages 35-40.

Fox, R.E. (1982a), "MRP, Kanban, or OPT What's best?", Inventories & Production Magazine, July-August.

Fox, R.E. (1982b), "OPT An Answer for America Part 2", Inventories & Production Magazine, November-December.

Fox, R.E. (1983a), “OPT An Answer for America Part 3”, Inventories & Production Magazine, January-February.

Fox, R.E. (1983b), “OPT An Answer for America Part 4”, Inventories & Production Magazine, March-April.

Fox, R.E. (1983c), “OPT vs. MRP - Thoughtware vs. Software Part 1”, Inventories & Production Magazine, November-December.

Fox, R.E. (1984), “OPT vs. MRP - Thoughtware vs. Software Part 2”, Inventories & Production Magazine, January-February, Pages 13-18.

Friedman, A.L. and Cornford D.S. (1989), “Computer Systems Development: History, Organisation and Implementation”, John Wiley and Sons, Chichester, UK.

Frisse, M. (1988), “From Text to Hypertext”, Byte, Vol. 13, No. 10, Pages 247-253.

Fryer, B. (1996), “Intranet mania”, DATAMATION, Vol. 42, No. 12, Pages 54-57.

George, C.S. (1968), “The History of Management Thought”, Prentice Hall, Hemel Hempstead, UK.

Goldratt, E.M. and Cox, J. (1993), “The Goal: A Process of On-Going Improvement”, Ed. 2, Gower Publishing Company, Aldershot, UK.

Goldratt, E.M. and Fox, R.E. (1985), “The Race”, Creative Output Inc.

Golhar, D.Y. and Stamm, C.L. (1991), “The Just in Time Philosophy: A literature review”, The International Journal of Production Research, Vol. 29, No. 4, Pages 657-676.

Goodwins, R. (1999), “Fast Forward”, PC Magazine, Ziff-Davis, February, Page 50.

Greek, D. (1996), “Insider Information”, Professional Engineering, 5 June, Pages 17-18.

Greek, D. (1997), "The Networked Factory", Professional Engineering, 9 July, Vol. 10, No. 13, Pages 24-25.

Greek, D. (2000a), "Whip Hand", Professional Engineering, 24 May, Vol. 13, No. 10, Page 43.

Greek, D. (2000b), "Communication Breakthrough", Professional Engineering, 21 June, Vol. 13, No. 12, Pages 31-32.

Greek, D. (2000c), "Net Profits", Professional Engineering, 5 July, Vol. 13, No. 13, Pages 33-34.

Grossman, W. (1998), "The End is Nigh", The Daily Telegraph Connected, 12 February, Page 13.

Hall, A.D. (1962), "A Methodology For Systems Engineering", Van Nostrand, Princeton, USA.

Hammer, M. and Champy, J. (1999), "Re-Engineering the Corporation: A Manifesto for Business Revolution", Brealey Publishing, London, UK.

Harrison, D.K., Primrose, P.L. and Leonard, R. (1986), "Identifying the Financial Advantages of Linking Computer Systems Within Manufacturing Companies", International Journal of Advanced Manufacturing Technology, Vol. 1, No. 2, Pages 5-15.

Harrison, D.K., Petty, D.J. and Leonard, R. (1989), "Criteria for assessing data transfer methods for CIM systems in practical engineering situations", Computer-Integrated Manufacturing Systems, Vol. 2, No. 4, Pages 228-235.

Harrison, D.K., Petty, D.J. and McLeod, C. (1995), "The Design of Integrated Manufacturing Information Systems", Proceedings of the Twelfth Conference of the Irish Manufacturing Committee, IMC12, on Competitive Manufacturing, 6-8 Sept, Pages 935-942.

Hassab, J.C. (1997), “Systems Management”, People, Computers, Machines, Materials, CRC Press, New York, USA.

Hendley, A. (1997), “An Introduction to Document Management”, iTx Marketing Services Ltd., Worcester UK.

Hogan, D. (1996), “Intranets”, Microsoft Advantage, August / September, Pages 59-61.

Hoyle, D. (1996), “Quality Systems - a new perspective”, Quality World, Vol. 22, No. 10, Pages 710-713.

Hussain, D.S. and Hussain, K.M. (1995), “Information Systems for Business”, Prentice Hall, Hemel Hempstead, UK.

Imai, M. (1986), “Kaizen - The Key to Japan’s Competitive Success”, McGraw-Hill, USA.

Inman, P. (1996), “The Intranet Threat to Lotus Notes”, BusinessAge, May, Pages 54-56.

Jacobson, I. (1995), “Formalizing Use-Case Modelling”, Journal of Object Oriented Programming, Vol. 8, No. 3, Pages 10-14.

Jacobson, I. and Christerson, M. (1995), “A Growing Consensus on Use Cases”, Journal of Object-Oriented Programming, Vol. 8, No. 1, Pages 15-19.

Jacobson, I., Griss, M. and Jonsson, P. (1997), “Making the Reuse Business Work”, (Reprinted from Software Reuse Architecture, Process and Organizations for Business Success, 1997), COMPUTER, Vol. 30, No. 10, Pages 36-42.

James, B. (1989), “The Systems Analysis Interview, A Guide to Interviewing in Organisations”, NCC Blackwell, Oxford, UK.

Jenkins, G.M. (1969a), "A Systems Study of a Petrochemical Plant", The Journal of Systems Engineering, Vol. 1, No. 1, Pages 90-101.

Jenkins, G.M. (1969b), "The Systems Approach", The Journal of Systems Engineering, Vol. 1, No. 1, Pages 3-49.

Kaarela, K., Oksanen, J. and Takalo, J. (1995), "An information model as a basis for hypermedia-based plant documentation", Computer Networks and ISDN Systems 27, Pages 751-764.

Kamenetzky, R.D. (1985), "Successful MRPII Implementation Can Be Complemented By Smart Scheduling, Sequencing Systems", Industrial Engineering, Oct, Pages 44-52.

Kautz, K. and McMaster, T. (1994), "The failure to introduce system development methods: a factor-based analysis", In: Levine, L. (Ed), Diffusion, Transfer and Implementation of Information Technology, Elsevier Science, Amsterdam, North Holland, Pages 275-287.

Kempner, T. (1987), The Penguin Management Handbook, Penguin Books, Harmondsworth.

Kenworthy, J. (1994), "Has Finite Scheduling Come of Age?", BPICS Control, October/November, Pages 26-27.

King, W.R. (1995), "Creating a Strategic Capabilities Architecture", Information Systems Management, Vol. 12, No. 3, Pages 67-69.

Kliem, R. (1984), "Writing Technical Procedures", Journal of Systems Management, Vol. 35, No. 10, Pages 30-33.

Knowles, R. (1995), "Electronic document management systems: how to select and implement the right documentation management system for your organisation", The Electronic Library, Vol. 13, No. 6, Pages 527-532.

LaLonde, W. and Pugh, J. (1997), “Internet/Intranet applications”, Journal of Object-Oriented Programming, Vol. 10, No. 2, Pages 63-67.

Lehner, F. (1993), “Quality Control in Software Documentation Based on Measurement of Text Comprehension and Text Comprehensibility”, Information Processing & Management, Vol. 29, No. 5, Pages 551-568.

Little, D. and Yusof, Y.Y. (1996), “Integrating MRPII Systems within the manufacturing business”, Advanced Manufacturing Processes Systems and Technologies (AMPST 96), No. 77, Pages 505-513.

Lundrigan, R. (1986), “What is this thing called OPT?”, Production and Inventory Management, Second Quarter, Pages 2-11.

McCarthy, S.P. (1996), “Intranet Technology A Growing Manufacturing Resource”, http://www.manufacturing.net/closeup/intranet.htm, December.

McKean, R.N. (1958), “Efficiency in Government Through Systems Analysis”, John Wiley and Sons Inc., New York, USA.

McLeod, C., Harrison, D.K. and Petty, D.J. (1995), “A Design Paradigm for Information Systems in Manufacturing Companies”, ICAM ‘95 Proceedings, 12-14 September.

McManus, J.J. (1987), “Control in the OPT Environment”, Production Engineer, July/August, Pages 20-21.

Meckler, A. (1997), “Java and Inter-Applet Communication”, Dr Dobb’s Journal, October, Pages 46 – 53.

Meleton, M.P. (Jr) (1986), “OPT - Fantasy or Breakthrough?”, Production and Inventory Management, Second Quarter, Pages 13-21.

Microsoft Encarta (1997), “Microsoft Encarta 98 Encyclopedia”, Microsoft Corporation, Redmond, USA.

Mills, H.D., Dyer, M., Linger, R.C. (1987), "Cleanroom Software Engineering", IEEE Software, September, Pages 19-24.

Minerva Industrial Systems, "Web-Site", http://www.minerva-is.com.

Mirel, B. (1988), "The Politics of Usability: The Organizational Functions of an In-House Manual", In: Doheny-Farina, S. (Ed), "Effective Documentation: What We Have Learned from Research", MIT Press, Cambridge Massachusetts, USA, No. 1, Pages 277-297.

Monden, Y. (1983), "Toyota Production System - Practical Approach to Production Management", Industrial Engineering and Management Press, Norcross, Georgia, USA.

Neely, A. (1997), "A practical approach to Defining Key Indicators", Measuring Business Excellence, Vol. 1, No. 1, Pages 42-46.

Netscape Communication Corp (1996), "Intranets Redefine Corporate Information Systems", http://home.netscape.com/comprod/at_work/white_paper/indept.html.

Nookabadi, A.S. and Middle, J.E. (1996), "A Generic IDEF0 Model of Quality Assurance Information Systems for the Design-to-Order Manufacturing Environment", IEEE Transactions on Components, Packaging and Manufacturing Technology, Part C, Vol. 19, No. 2, Pages 78-84.

Orlicky, J.A. (1975), "Material Requirements Planning: The New Way of Life in Production and Inventory Management", McGraw-Hill Book Company, USA.

Petty, D.J. and Harrison, D.K. (1995), "Implementation of a Business Control System in a Divisionalised Manufacturing Company - A Case Study", Proceedings of the Eleventh National Conference on Manufacturing Research, Leicester, 12-14 September, Pages 331-335.

Porter, K. and Little, D. (1996), "Finite Scheduling and MRP - Revealing Relationships", BPICS Control, May, Pages 23-26.

Porter, M.E. and Millar, V.E. (1985), "How information gives you competitive advantage", Harvard Business Review, July-August, Pages 149-160.

Pullin, J. (2000a), "The Best Laid Plans", Professional Engineering, 19 April, Vol. 13, No. 8, Page 34.

Pullin, J. (2000b), "Chain Reactions", Professional Engineering, 24 May, Vol. 13, No. 10, Pages 34-39.

Ramey, J. (1988), "How People Use Computer Documentation: Implications for Book Design", In: Doheny-Farina, S. (Ed), "Effective Documentation: What We Have Learned from Research", MIT Press, Cambridge Massachusetts, USA, No. 7, Pages 143-158.

Reynolds, K.A. and Wainwright, C.E.R. (1997), "Information Integration : A Survey", Proceedings of the 13th NCMR Conference, Glasgow Caledonian University, 9-11 September, Pages 590-595.

Reynolds, K.A., Winchester, M.J. and Wainwright, C.E.R. (1997), "A Methodology to Aid the Development of Information Technology Strategies", Proceedings 5th International IEE Factory 2000 Conference, Cambridge University, UK, April, Pages 429-436.

Rosman, G., van der Meer, K. and Sol, H.G. (1996), "The Design of Document Information Systems", Journal of Information Science, Vol. 22, No. 4, Pages 287-297.

Rothery, B. (1995), "ISO14000 and ISO9000", Gower, Aldershot, Hampshire, UK.

Rousseau, B., Ruggier, M. and Smith, M. (1994), "An Automated System for the Maintenance of Multiform Documentation", International Journal of Modern Physics C, Vol. 5, No. 5, Pages 767-768.

Roy, R. and Meikle, S.E. (1995), "The Role of Discrete Event Simulation Techniques in Finite Capacity Scheduling", Journal of the Operational Research Society, No. 46, Pages 1310-1321.

Rubens, B.K. (1988), "Similarities and Differences in Developing Effective Online and Hardcopy Tutorials", In: Doheny-Farina, S. (Ed), "Effective Documentation: What We Have Learned from Research", MIT Press, Cambridge Massachusetts, USA, No. 8, Pages 159-184.

SAP, "Web-Site", http://www.sap-ag.de.

Saunders, M.N.K., Lewis, P. and Thornhill, A. (2000), "Research Methods for Business Students", Second Edition, Pearson Education Limited, London, UK.

Schonberger, R.J. (1984), "Just In Time Production Systems: Replacing Complexity With Simplicity In Manufacturing Management", Industrial Engineering, Vol. 16, No. 10, Pages 52-63.

Segerstedt, A. (1996), "Formulas of MRP", International Journal of Production Economics, Vol. 46-47, Pages 127-136.

Sherer, S.W., Kouchakdjian, A. and Arnold, P.G. (1996), "Experience using cleanroom software engineering", IEEE Software, Vol. 13, No. 3, Pages 69-76.

Shingo, S. (1985), "A Revolution In Manufacturing: The SMED System", Productivity Press, Portland, USA.

Sirisawat, S. and Duffill, A.W. (1997), "Intranet in Manufacturing", Proceedings of the 13th NCMR Conference, Glasgow Caledonian University, 9-11 September, Pages 547-552.

Skinner, W. (1974), "The Focused Factory", Harvard Business Review, May-June, Pages 113-121.

Small, C., Baines, R.W, Colquhoun, G.J. (1997a), "A structured view of change projects in three UK manufacturing companies", Proceedings of the 1st International Conference on Manufacturing Enterprises - Stakeholders, Engineering, Logistics and Achievement (MeSELA), Loughborough University, Pages 197-202.

Small, C., Baines, R.W, Colquhoun, G.J. (1997b), "The application of graphical modelling methods in manufacturing change", Proceedings of the 13th NCMR Conference, Glasgow Caledonian University, 9-11 September, Pages 553-558.

Softworld Report and Directory (1998), "Supply Chain Software Solutions", Interactive Information Services Ltd., Surrey, UK.

Stroud, K.A. (1992), "Engineering Mathematics Programmes and Problems", Edition 3, Macmillan Education Ltd., Basingstoke, UK.

Swamson, E.B. (1982), "A View of Information System Evolution", In Hawgood, J. (Editor), Evolutionary Information Systems, North-Holland Publishing Company, Proceedings of the IFIP TC 8 Working Conference on Evolutionary Information Systems, Pages 55-63.

Swann, D. (1986), "Using MRP for Optimized Schedules (Emulating OPT)", Production and Inventory Management, Second Quarter, Pages 30-37.

T&N PLC, "Annual Reports and Accounts 1996", UK.

Tabor, S.W., Pryor, A.N. and Gutierrez, C.F. (1997), "Improving Corporate Communications with Intranets", Information Strategy, Vol. 14, No. 1, Pages 7-12.

Tian, G.Y., Zhao, Z.X. and Baines, R.W. (1997), "Agile manufacturing information based on computer network of the World-Wide Web (WWW)", Proceedings of the 13th NCMR Conference, Glasgow Caledonian University, 9-11 September, Pages 571-575.

Tranmer, J. (1996), "Overcoming the problems to integrated management systems", Quality World, Vol. 22, No. 10, Pages 714-718.

Tuckman, B.W. (1978), "Conducting Educational Research", 2nd Edition, Harcourt Brace Jovanovich, Inc, USA.

Urmson, J.O., and Ree, J. (Editors) (1993), "The Concise Encyclopedia of Western Philosophy and Philosophers", Routledge, London, UK.

VDA (1996), "Quality Systems Audit: Based on ISO 9001 and ISO 9004", Part 1, 3rd Revised Edition, Druckerei Henrich GmbH, Germany.

Vollmann, T.E., Berry, W.L. and Whybark, C.D. (1997), "Manufacturing Planning and Control Systems", Edition 4, McGraw Hill, USA.

Wallace, T.F. (1990), "MRPII Making It Happen - The Implementer's Guide to Success with Manufacturing Resource Planning", Edition 2, Oliver Wight Ltd. Publications, Essex Junction, USA.

Walton, M. (1994), "The Deming Management Method", Management Books 2000 Ltd., Oxon, UK.

Ward, J. and Griffiths, P. (1997), "Strategic Planning for Information Systems", Edition 2, John Wiley and Sons, Chichester, UK.

Waters, D. (1997), "Quantitative Methods for Business", 2nd Edition, Addison Wesley Longman Publishers Ltd, Harlow, UK.

Watson, K. (1996), "Finite scheduling - who needs IT", The Engineer, March, Page 17.

Webopedia (2000), "Fourth-Generation Languages", http://webopedia.internet.com /TERM/f/fourth_generation_language.htm.

Wight, O.W. (1984), "Manufacturing Resource Planning: MRPII: Unlocking Americas Productivity Potential", Revised Edition, Oliver Wight Ltd. Publications, Essex Junction, USA.

Wilson, B. (1990), "Systems: Concepts, Methodologies and Applications", 2nd Edition, John Wiley and Sons, Chichester, UK.

Wilson, F., Desmond, J. and Roberts, H. (1994), "Success and Failure of MRP II Implementation", British Journal of Management, Vol. 5, No. 3, Pages 221-240.

Wood, G.C. and Leonard, R. (1978), "The Design and Implementation of a Flexible, University-Wide, Total Technology Programme", UMEE, Vol. 6, No. 4, Pages 289-292.

Wortman, J.C., Euwe, M.J., Taal, M. and Wiers, V.C.S. (1996), "A review of capacity planning techniques within standard software packages", Production Planning and Control, Vol. 7, No. 2, Pages 117-128.

Wu, B. (1992), "Manufacturing Systems Design and Analysis", Chapman & Hall, London, UK.

Young, J. (1996), "Object Lesson", Forbes, August 26, Pages 80-81.