This item is held in Loughborough University’s Institutional Repository (https://dspace.lboro.ac.uk/) and was harvested from the British Library’s EThOS service (http://www.ethos.bl.uk/). It is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to: http://creativecommons.org/licenses/by-nc-nd/2.5/
Production Data Analysis for Discrete Component Manufacture

By

Richard William Bagshaw

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of

Doctor of Philosophy of Loughborough University

September 1999

© by Richard William Bagshaw, 1999
ABSTRACT

This thesis reports research into a workshop oriented PC-based machine and inspection facility for a contemporary metalworking SME. It identifies a production data analysis framework, which is supported by the use of order and manufacturing models. A major feature of the framework is the ability to produce rapid manufacturing control through feedback data from both the inspection and manufacturing data analysis activities in order to influence the responsiveness of manufacturing disturbances experienced through the machining of discrete prismatic components. The major contribution of this thesis explores a production data analysis framework, which forms the basis of a prototype computational facility that closes the quality information feedback loop void that exists within manufacturing.

The novel approach employed by the production data analysis framework provides both product and manufacturing process control and involves a number of phases in order to close the manufacturing feedback loop. These phases are described and involve the concurrent machine operation and inspection planning, simultaneous production code generation, comparative tolerance analysis and manufacturing data analysis of prismatic components. The information requirements of both the order and manufacturing models to support the functionality of each phase of the production data analysis framework are also examined and discussed.

An integrated multi-functional prototype production data analysis software tool supported by information models has been developed for a limited number of manufacturing features. This software tool has been tested through the application of a case study and has proven the production data analysis methodology to be of strong potential for use within a CAE environment.
ACKNOWLEDGEMENTS

I wish to express my sincere thanks to my supervisor, Dr S.T. Newman, for his supervision, encouragement, interest and friendship in this research. I am also indebted to Professor R. Bell for his help, support and advice throughout the duration of this work. I am grateful to my colleagues, Dr S. Rahimifard, and Dr K.T.K Toh who have been involved with the SME Research Group, for the assistance inspiring discussions.

I would also like to thank all my friends for their friendship, guidance and inspiration throughout the period of this work; in particular: Mr D. Jeffreys, Mr D. Walters, Mr C. Turner, Mr T.W. Downham, Mr D.W. Hurrell and Mr J Singh of the department.

I would also like to express my deepest gratitude to Mr G. Hawkesford, Mr M. Pickering, of CAMTEK UK Ltd and Mr M. Goldfeld of, CAMTEK Pacific Ltd for the encouragement and technical support.

I wish to acknowledge my appreciation to Professor D.J. Williams for giving myself the opportunity to undertake this work, and to both the EPSRC Postgraduate Studentship Award and the EPSRC (GR/L27077) “IT Tools to Improve the Manufacturing Performance of Metalworking SMEs” research programme for providing the funding.

Last but not least I wish to express my deepest gratitude to Mr M. Bagshaw and Mrs M.A. Geeson for their unwavering support and encouragement during this period of research.
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>AIS</td>
<td>Application Interface Specification</td>
</tr>
<tr>
<td>ANC</td>
<td>Advanced Numerical Control</td>
</tr>
<tr>
<td>ANS</td>
<td>Artificial Neural Systems</td>
</tr>
<tr>
<td>APT</td>
<td>Automatically Programmed Tool</td>
</tr>
<tr>
<td>B-Rep</td>
<td>Boundary Representation</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer-Aided Design</td>
</tr>
<tr>
<td>CADD</td>
<td>Computer Aided Design and Drafting</td>
</tr>
<tr>
<td>CAE</td>
<td>Computer Aided Engineering</td>
</tr>
<tr>
<td>CAI</td>
<td>Computer Aided Inspection</td>
</tr>
<tr>
<td>CAIDA</td>
<td>Computer-Aided Inspection Data Analyser</td>
</tr>
<tr>
<td>CAM</td>
<td>Computer Aided Manufacture</td>
</tr>
<tr>
<td>CAM-I</td>
<td>Computer-Aided Manufacturing – International</td>
</tr>
<tr>
<td>CAPP</td>
<td>Computer-Aided Process Planning</td>
</tr>
<tr>
<td>CE</td>
<td>Concurrent Engineering</td>
</tr>
<tr>
<td>CIM</td>
<td>Computer Integrated Manufacturing</td>
</tr>
<tr>
<td>CLDD</td>
<td>Composite Logical Data Design</td>
</tr>
<tr>
<td>CLIPS</td>
<td>C Language Integrated Production System</td>
</tr>
<tr>
<td>CMM</td>
<td>Co-ordinate Measuring Machine</td>
</tr>
<tr>
<td>CNC</td>
<td>Computer Numerical Control</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CSG</td>
<td>Constructive Solid Geometry</td>
</tr>
<tr>
<td>CVM</td>
<td>CMM Verification Module</td>
</tr>
<tr>
<td>D&amp;T</td>
<td>Dimensioning and Tolerancing</td>
</tr>
<tr>
<td>DCC</td>
<td>Direct Computer Control</td>
</tr>
<tr>
<td>DD</td>
<td>First Cut Design</td>
</tr>
<tr>
<td>DFD</td>
<td>Data Flow Diagrams</td>
</tr>
<tr>
<td>DFM</td>
<td>Design for Manufacture</td>
</tr>
<tr>
<td>DME</td>
<td>Dimensional Measuring Equipment</td>
</tr>
<tr>
<td>DMIS</td>
<td>Dimensional Measuring Interface Standard</td>
</tr>
<tr>
<td>DTAIS</td>
<td>Dimensioning and Tolerancing Application Interface Specification</td>
</tr>
<tr>
<td>EAD</td>
<td>External Access Directions</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>ECM</td>
<td>Electro-Chemical Machining</td>
</tr>
<tr>
<td>ELH</td>
<td>Entity Life Histories</td>
</tr>
<tr>
<td>EPS-1</td>
<td>Expert Programming System – One</td>
</tr>
<tr>
<td>EPSRC</td>
<td>Engineering and Physical Science Research Council</td>
</tr>
<tr>
<td>ESP</td>
<td>Experimental Solids Proposal</td>
</tr>
<tr>
<td>ESPRIT</td>
<td>European Strategic Program for Research in Information Technology</td>
</tr>
<tr>
<td>EXCAP</td>
<td>Expert Computer Aided Planner</td>
</tr>
<tr>
<td>FFBP</td>
<td>Feed Forward Back Propagation</td>
</tr>
<tr>
<td>FMS</td>
<td>Flexible Manufacturing Systems</td>
</tr>
<tr>
<td>GAC</td>
<td>Global Accessibility Cone</td>
</tr>
<tr>
<td>GD&amp;T</td>
<td>Geometric Dimensioning and Tolerancing</td>
</tr>
<tr>
<td>GT</td>
<td>Group Technology</td>
</tr>
<tr>
<td>HTBasic</td>
<td>High Tech Basic</td>
</tr>
<tr>
<td>ICF</td>
<td>Inspection Code Fragments</td>
</tr>
<tr>
<td>IDEEAA</td>
<td>Intelligent Design Environment for Engineering Applications</td>
</tr>
<tr>
<td>IDEF0</td>
<td>ICAM Definition Level 0</td>
</tr>
<tr>
<td>IDEF1x</td>
<td>ICAM Definition Level 1x</td>
</tr>
<tr>
<td>IGES</td>
<td>Initial Graphics Exchange Specification</td>
</tr>
<tr>
<td>IMPPACT</td>
<td>Integrated Modelling of Product and Processes using Advanced Computer Technologies</td>
</tr>
<tr>
<td>IMS</td>
<td>Integrated Metrology Systems</td>
</tr>
<tr>
<td>INSPEX</td>
<td>Inspection Expert Planner</td>
</tr>
<tr>
<td>IOP</td>
<td>Index of Priority</td>
</tr>
<tr>
<td>IPF</td>
<td>Inspection Plan Fragments</td>
</tr>
<tr>
<td>IPPEX</td>
<td>Inspection Process Planning Expert</td>
</tr>
<tr>
<td>ISS</td>
<td>Information Support System</td>
</tr>
<tr>
<td>JESS</td>
<td>Java Expert System Shell</td>
</tr>
<tr>
<td>LAC</td>
<td>Local Accessibility Cone</td>
</tr>
<tr>
<td>LDD</td>
<td>Logical Dialogue Design</td>
</tr>
<tr>
<td>LDST</td>
<td>Logical Data Structuring Technique</td>
</tr>
<tr>
<td>LMC</td>
<td>Least Material Condition</td>
</tr>
<tr>
<td>LUT-FBDS</td>
<td>Loughborough University of Technology – Feature Based Design System</td>
</tr>
<tr>
<td>MCG</td>
<td>Manufacturing Code Generator</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>MDA</td>
<td>Manufacturing Data Analysis</td>
</tr>
<tr>
<td>MMC</td>
<td>Maximum Material Condition</td>
</tr>
<tr>
<td>MOSES</td>
<td>Model Oriented Simultaneous Engineering System</td>
</tr>
<tr>
<td>MTI</td>
<td>Machine Tool Interface</td>
</tr>
<tr>
<td>NC</td>
<td>Numerical Control</td>
</tr>
<tr>
<td>OOIP</td>
<td>Object-Oriented Inspection Planner</td>
</tr>
<tr>
<td>PAD</td>
<td>Probe Approach Direction</td>
</tr>
<tr>
<td>PADDES</td>
<td>Production data Analysis Distributed Diagnostic Expert System</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PDC</td>
<td>Physical Design Control</td>
</tr>
<tr>
<td>PDDI</td>
<td>Product Definition Data Interface. A USAF Project</td>
</tr>
<tr>
<td>PDES</td>
<td>Product Data Exchange Standard</td>
</tr>
<tr>
<td>PEPS</td>
<td>Production Engineering Productivity System</td>
</tr>
<tr>
<td>PEPSIM</td>
<td>Production Engineering Productivity System inspection Module</td>
</tr>
<tr>
<td>PME</td>
<td>Probe Movement Envelope</td>
</tr>
<tr>
<td>PO</td>
<td>Process Outlines</td>
</tr>
<tr>
<td>PS</td>
<td>Program Specifications</td>
</tr>
<tr>
<td>RASOR</td>
<td>Rules And Systems Of Rules</td>
</tr>
<tr>
<td>RDA</td>
<td>Relational Data Analysis</td>
</tr>
<tr>
<td>SADT</td>
<td>Structured Analysis and Design Technique</td>
</tr>
<tr>
<td>SAG</td>
<td>Surface Adjacency Graph</td>
</tr>
<tr>
<td>SE</td>
<td>Simultaneous Engineering</td>
</tr>
<tr>
<td>SET</td>
<td>Standard d'Exchange et Transfer. French Data exchange standard</td>
</tr>
<tr>
<td>SME</td>
<td>Small Manufacturing Enterprise</td>
</tr>
<tr>
<td>SPC</td>
<td>Statistical Process Control</td>
</tr>
<tr>
<td>SQC</td>
<td>Statistical Quality Control</td>
</tr>
<tr>
<td>SSADM</td>
<td>Structured System Analysis and Design Methodology</td>
</tr>
<tr>
<td>STD</td>
<td>State Transition Diagrams</td>
</tr>
<tr>
<td>STEP</td>
<td>Standard for the Transfer and Exchange of Product Model Data</td>
</tr>
<tr>
<td>TMS</td>
<td>Tolerance Modelling System</td>
</tr>
<tr>
<td>TOPAC</td>
<td>Tool Path Correction Algorithm</td>
</tr>
<tr>
<td>TPME</td>
<td>Total Probe Movement Envelope</td>
</tr>
<tr>
<td>TQM</td>
<td>Total Quality Management</td>
</tr>
<tr>
<td>UMIST</td>
<td>University of Manchester Institute of Science and Technology</td>
</tr>
<tr>
<td>VDA/FS</td>
<td>Verband des Automobilindustrie Flachen Schnittstelle. A German extension of IGES for the transfer of surface data</td>
</tr>
<tr>
<td>---------</td>
<td>----------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>XBF</td>
<td>Experimental Boundary File</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

Abstract
Acknowledgements
Abbreviations
Table of Contents

1.0 Introduction 1
1.1 Introduction 1
1.2 The Structure of the Thesis 2

2.0 Scope of Research 6
2.1 Introduction 6
2.2 Research Objectives 6
2.3 Scope of Research 7
2.4 Production Data Analysis Framework 7
  2.4.1 Machine and Inspection Planning of Discrete Component Parts 7
  2.4.2 Production Code Generation 8
  2.4.3 Geometric Error Determination by Comparative Tolerance Analysis of Inspection Results 8
  2.4.4 Production Error Diagnosis through the Application of Manufacturing Data Analysis 8
  2.4.5 Information Data Resource Integration 8
  2.4.6 Design of Practical Experiments 9

3.0 Product Information Modelling for Manufacture 10
3.1 Introduction 10
3.2 Geometric Modelling 10
  3.2.1 Wire Frame Modelling 11
  3.2.2 Surface Modelling 11
  3.2.3 Solid Modelling Techniques 11
3.3 Feature Based Modelling 13
  3.3.1 Feature Taxonomies 14
3.3.2 Human Assisted Feature Definitions 15
3.3.3 Automatic Feature Recognition 15
3.3.4 Design by Features 16

3.4 Machine Operation Planning & Sequencing 16
3.4.1 The Variational Approach to Process Planning 17
3.4.2 The Generative Approach to Process Planning 18
3.4.3 Machine Operation Planning 18
3.4.4 Cutting Tool Selection 19
3.4.5 Tool Path Simulation 20
3.4.6 NC Part Program Generation 21

3.5 Fundamentals of Simultaneous Engineering 21
3.5.1 Product Modelling Concept 22
3.5.2 Integrated Product Modelling Environments 25
3.5.3 Data Transfer Standard Formats 28
  3.5.3.1 Initial Graphics Exchange Standards (IGES) 29
  3.5.3.2 The Product Definition Data Interface (PDDI) 30
  3.5.3.3 Product Data Exchange Standard (PDES) 31
  3.5.3.4 ISO Standardisation and the Standard for the Transfer 32
  and Exchange of Product Model Data (STEP)
  3.5.3.5 Dimensional Measuring Interface Specification (DMIS) 34
3.5.4 Critique - Simultaneous Engineering Approaches 34

3.6 Order Model Concept 36
3.6.1 Critique - Order Modelling 37

3.7 Manufacturing Resource Modelling 37
3.7.1 Critique - Manufacturing Modelling 38

3.8 Information Modelling Methodologies 39
3.8.1 Critique - Information Modelling Techniques 42

4.0 Review of Contemporary Automated Inspection Research 43
4.1 Introduction 43
4.2 Tolerancing Methods and Standards

4.2.1 British Standards Institution's BS308: Engineering Drawing Practice

4.2.2 American National Standards Institute: Dimensioning and Tolerancing

4.2.3 The Importance of Following Geometric, Dimensioning and Tolerancing Specifications

4.2.4 The Addition of Tolerance Information to Solid Models

4.2.5 Tolerance Modelling

4.2.6 Critique – Tolerancing Methods and Standards

4.3 Inspection Planning for CMMs

4.3.1 Current CMM Operating Practice

4.3.2 Feature Recognition and Inspection Requirements Specification

4.3.3 Measurement Planning and Sequencing

4.3.4 Probe Configuration and Orientation Selection

4.3.5 Probe point and Path Planning

4.3.6 Inspection Plan Simulation and Verification

4.3.7 Inspection Plan Execution

4.3.8 Critique – Inspection Planning for CMMs

4.4 Inspection Results Evaluation

4.4.1 Requirements for Research in CAD Based Automated Inspection

4.4.2 CAD/CMM Interfacing

4.4.3 Integration through Data Exchange Standards

4.4.4 Inspection Integration into Automated Manufacturing

4.4.5 Automated Inspection Environments

4.4.6 Inspection Based on CAD Models

4.4.7 Emerging Research Issues in the Context of CAD-directed Inspection

4.4.8 Critique – Inspection Results Evaluation

4.5 Current State of Commercial Inspection Systems

4.5.1 Automation Software's PC-DMIS

4.5.2 International Metrology System's Virtual-DMIS

4.5.3 LK's Visual CMES
5.0 Manufacturing Fault Diagnosis Approaches 80
5.1 Introduction 80
5.2 Traditional Troubleshooting Concepts 81
  5.2.1 Checklist Approach 81
  5.2.2 What Changed Approach 82
  5.2.3 Morphological approach 82
  5.2.4 Brainstorming 82
  5.2.5 Weighted Factor Analysis 83
  5.2.6 Quality Circles 83
  5.2.7 Relevance Trees 84
  5.2.8 Critique - Traditional Troubleshooting Concepts 84
5.3 Structured Troubleshooting Approaches 84
  5.3.1 Statistical Process Control (SPC) 84
  5.3.2 Expert System Paradigm 86
    5.3.2.1 Expert System Advantages within a Diagnostic Context 88
    5.3.2.2 The Representation of Knowledge 89
    5.3.2.3 Knowledge Acquisition and the Expert System's Development Cycle 91
  5.3.3 Inference Strategies and Methods 93
  5.3.4 Expert Systems Applied to other Problem Domains 95
  5.3.5 Neural Networks 96
  5.3.6 Critique - Structured Troubleshooting Approaches 98
5.4 Machine Tool Error Diagnosis 99
  5.4.1 Manufacturing Error Classification 99
  5.4.2 Condition Monitoring of Manufacturing Machine Tools 100
  5.4.3 Manufacturing Data Analysis of Production Errors 103
  5.4.4 Critique - Machine Tool Error Diagnosis 107
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.0</td>
<td>The Production Data Analysis Research</td>
<td>108</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>108</td>
</tr>
<tr>
<td>6.2</td>
<td>EPSRC (GR/L27077) Project: IT Tools to Improve the Manufacturing</td>
<td>108</td>
</tr>
<tr>
<td></td>
<td>Performance of Metalworking SMEs</td>
<td></td>
</tr>
<tr>
<td>6.2.1</td>
<td>The Contemporary Small Manufacturing Enterprise</td>
<td>109</td>
</tr>
<tr>
<td>6.2.2</td>
<td>The SME Reference Model</td>
<td>111</td>
</tr>
<tr>
<td>6.3</td>
<td>The Production Data Analysis Framework</td>
<td>111</td>
</tr>
<tr>
<td>6.3.1</td>
<td>Machine and Inspection Planning</td>
<td>112</td>
</tr>
<tr>
<td>6.3.2</td>
<td>Production Code Generation</td>
<td>113</td>
</tr>
<tr>
<td>6.3.3</td>
<td>Comparative Tolerance Analysis</td>
<td>114</td>
</tr>
<tr>
<td>6.3.4</td>
<td>Manufacturing Data Analysis</td>
<td>114</td>
</tr>
<tr>
<td>6.3.5</td>
<td>Data Resource Model Integration</td>
<td>115</td>
</tr>
<tr>
<td>6.4</td>
<td>Operational Structure of the Production Data Analysis Framework</td>
<td>115</td>
</tr>
<tr>
<td>7.0</td>
<td>The Production Data Generation and Inspection Analysis</td>
<td>118</td>
</tr>
<tr>
<td>7.1</td>
<td>Introduction</td>
<td>118</td>
</tr>
<tr>
<td>7.2</td>
<td>Production Engineering Productivity System Inspection Module (PEPSIM)</td>
<td>118</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Manufacturing Feature-based Characterisation</td>
<td>119</td>
</tr>
<tr>
<td>7.3</td>
<td>Machine and Inspection Planning</td>
<td>120</td>
</tr>
<tr>
<td>7.3.1</td>
<td>PEPSIM – Tolerance Representation and Interpretation</td>
<td>121</td>
</tr>
<tr>
<td>7.3.2</td>
<td>PEPSIM – Feature Probing Point Distribution</td>
<td>123</td>
</tr>
<tr>
<td>7.3.3</td>
<td>PEPSIM – Feature Based Probing Path Generation</td>
<td>127</td>
</tr>
<tr>
<td>7.4</td>
<td>Operational Structure of the Machine and Inspection Planner</td>
<td>129</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Create Part Model Geometry</td>
<td>129</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Create Operation Plan</td>
<td>131</td>
</tr>
<tr>
<td>7.4.3</td>
<td>Create Inspection Procedure</td>
<td>132</td>
</tr>
<tr>
<td>7.5</td>
<td>Production Code Generation</td>
<td>133</td>
</tr>
<tr>
<td>7.6</td>
<td>Operational Structure of the Production Code Generator</td>
<td>135</td>
</tr>
<tr>
<td>7.7</td>
<td>Comparative Tolerance Analysis</td>
<td>138</td>
</tr>
<tr>
<td>7.8</td>
<td>Operational Structure of the Comparative Tolerance Analyser</td>
<td>145</td>
</tr>
<tr>
<td>7.8.1</td>
<td>Comparative Tolerance Analysis of Billet Feature</td>
<td>145</td>
</tr>
</tbody>
</table>
7.8.2 Comparative Tolerance Analysis of Construction Features 145
7.8.3 Output of Comparative Tolerance Analysis Results 147
7.9 Summary of Production Code Generation and Inspection Analysis 149

8.0 Manufacturing Data Analysis and Diagnosis 151
8.1 Introduction 151
8.2 Production Data Analysis Distributed Diagnostic Expert System (PADDES) 151
  8.2.1 Expert System Development Tool 151
  8.2.2 Component Fact Representation for PADDES 152
  8.2.3 Knowledge Acquisition for PADDES 153
  8.2.4 PADDES Manufacturing Error Categorisation 155
  8.2.5 Knowledge Representation of PADDES 156
  8.2.6 PADDES Consultation Documentation 162
8.3 Operational Structure of the Production Data Analysis Distributed Diagnostic Expert System (PADDES) 164
  8.3.1 Obtain Component Feature Fact File 164
  8.3.2 Obtain Machine Dependent Fault Library 164
  8.3.3 Conduct Manufacturing Data Analysis 167
8.4 Summary of Manufacturing Data Analysis and Diagnosis 168

9.0 Information Models to Support Production Data Analysis 169
9.1 Introduction 169
9.2 The SME Reference Model 169
9.3 The SME Order Sub-model 170
  9.3.1 SME Order Sub-Model to Support Production Data Analysis 172
    9.3.1.1 Nominal Part Attribute Aspect of the PDA Framework’s Order Sub-model 172
    9.3.1.2 Actual Feature Attribute Aspect of the PDA Framework’s Order Sub-model 176

xii
9.4 The SME Manufacturing Sub-model
9.4.1 SME Manufacturing Sub-model to Support Production Data Analysis

10.0 Production Data Analysis Test Cases
10.1 Introduction
10.2 Experimental Parts Design
10.3 Production Data Analysis Component Study
   10.3.1 Create Part Model Geometry
   10.3.2 Create Operation Plan
   10.3.3 Create Inspection Procedure
   10.3.4 Produce Production Code
   10.3.5 Comparative Tolerance Analysis
   10.3.6 Manufacturing Data Analysis
      10.3.6.1 Individual Feature MDA Diagnosis
      10.3.6.2 Combined Feature MDA Diagnosis
10.4 Test Case Summary

11.0 Concluding Discussions
11.1 Introduction
11.2 The Production Data Analysis Framework
11.3 Machine and Inspection Planning
11.4 Production Code Generation
11.5 Comparative Tolerance Analysis
11.6 Manufacturing Data Analysis (MDA)
11.7 Data Resource Model Integration

12.0 Conclusions and Further Work
12.1 Introduction
12.2 Conclusions
12.3 Further Work
   12.3.1 Extend Feature Taxonomy
12.3.2 Apply Methodology to other Component Configurations 230
12.3.3 Apply to In-process / Post-machining Inspection at Machine Tool 230
12.3.4 Extend Tolerance Analysis to Cover the Complete 230
  ASME Y14.5M Standard
12.3.5 Extension of the Manufacturing Data Analysis Knowledge Base 231
12.3.6 Apply the PDA Framework Methodology into a Commercially 231
  Available CAE System

References 232

Appendices 261
I PEPS 3.0 Milling Expert Part Programming System Manual 261
II Complete IDEF0 Representation of the Production Data Analysis 289
  Framework
III Decision Tree Knowledge Representation of PADDES 299
IV List of Author’s Publications 326
Chapter 1

INTRODUCTION

1.1 Introduction
With the reduction in costs and the increase in processing power of contemporary computer systems, the Computer Integrated Manufacturing (CIM) ideals which were exclusive to installations within large companies whose business activities are deterministic in nature, are now becoming viable for the smaller manufacturing enterprises (SMEs). However, in most instances these large inflexible applications cannot readily be scaled down and directly applied to the volatile manufacturing environment within which a contemporary metalworking SME must operate. With increasing emphasis being placed by the customer for improved product quality, the traditional non-adding value activity of inspection is now regarded as being essential to the survival of any manufacturing enterprise operating within today's global market.

The field of automated inspection has received a considerable amount of interest from both the academic and industrial communities over the last 20 years. These research interests have largely been concentrated on the inspection planning and geometric error determination aspects of the discipline within a limited application range, i.e. sculptured surfaces, cylindrical component configurations, etc. The plethora of experimental prototype facilities yielded from these research contributions are not only extremely complicated, computationally expensive, but also only concentrate on various aspects of the inspection activity, such as tolerance representation, path planning, collision detection, etc., at the expense of others namely, results analysis. The resultant prototypes become an additional island of automation that take no consideration of the requirement or data generated from other peer applications.

This research aims to bridge these islands of automation by integrating the machining and inspection planning and feedback analysis activities into a comprehensive framework and thus closing the manufacturing information feedback loop that currently exists between design/planning and manufacture. This framework, termed the Production Data Analysis (PDA) framework, although directed at the inter-
disciplinary and volatile working practices of small manufacturing enterprises (SMEs), can be equally applied to larger and more deterministic enterprises.

1.2 The Structure of the Thesis

The structure of the thesis is divided into four main sections of background/literature review, theoretical research, experimental research and research conclusions as depicted in figure 1.1. The background/literature review section comprises five chapters and provides a detailed review of pertinent research publications and background knowledge of the research. Chapter 1 provides the main introduction to the research contribution and outlines the structure of the thesis. The principal aims, objectives, scope of the research and the introduction to the production data analysis is presented in chapter 2. Chapter 3 provides a review of the methodologies employed for product information modelling and includes geometric and feature-based modelling, simultaneous engineering, order and manufacturing philosophies. Contemporary research into the automation of the inspection activities, ranging from inspection planning and execution on co-ordinate measuring machines to the analysis of the inspection results, from both industrial and academic institutions is described in chapter 4. The final chapter in the background/literature review section outlines the strategies and methodologies adopted by industrial and academic institutions for both manual and automated diagnosis of manufacturing errors from the analysis of sensory or inspection information. A pictorial view of this supporting literature and the associated relationships is shown in figure 1.2.

The theoretical research section of the thesis comprises four chapters, and commences with chapter 6 and identifies the context for which the research contribution is directed. The contemporary SME as identified by the EPSRC GR/L27077 research project is introduced and describes the mapping of the novel production data analysis concept onto this SME representation. Chapters 7, 8 and 9 detail the conceptualisation and realisation of the production data analysis framework with respect to machine and inspection planning, manufacturing error diagnosis and the supporting information model representation.

The experimental research section, chapter 10 describes a series of computer simulation experiments based upon a single test component which highlights the capabilities of the novel prototype production data analysis system. The experiments
are designed to systematically test every facet of the comprehensive prototype production data analysis system.

The final section of the thesis, namely research conclusions consists of two chapters. In chapter 11, concluding discussions analyses the wide range of research issues reported in this thesis from the initial statement of requirements and conceptualisation of the PDA framework to the realisation of an integrated and comprehensive production data generation and analysis system. The final chapter of the thesis provides a list of suggested research avenues that can be explored for the possible continuation of this research.

The appendices include a brief overview of the CAM software package employed as the foundation for the realisation of the prototype PDA system, the results of the activity investigation of the PDA framework, a comprehensive decision tree representation of the knowledge employed by the prototype PDA system for manufacturing error diagnosis, and a list of current author’s publications.
Figure 1.1 The Structure of the Chapters within the Thesis
Product Information Modelling for Manufacture (Chapter 3)

- Geometric modelling methods
  - Feature based modelling methodology
  - Design by features

- Machine operation planning & sequencing
  - Cutter tool selection
  - NC part program generation

Fundamentals of simultaneous engineering

- Product modelling concept
- Integrated product modelling environments
- Data transfer standard formats

Order model concept

- Manufacturing resource modelling
- Information modelling techniques

Review of Contemporary Automated Inspection Research (Chapter 4)

- Tolerance methods & standards
- Inspection planning for CMMs
- Current CMM operating practice
- Feature recognition & inspection requirements specification
- Probe configuration & orientation selection
- Probing point & path planning
- Inspection plan simulation & verification
- Inspection plan execution

Inspection result evaluation

- CAD/CMM interfacing
- Integration through data exchange standards
- Inspection integration into automated manufacturing
- Automated inspection environments
- Inspection based on CAD models
- Emerging research issues in the context of CAD-directed inspection
- Current state of commercial inspection systems

Figure 1.2 Production Data Analysis Framework Literature Topics
Chapter 2

THE SCOPE OF RESEARCH

2.1 Introduction
This chapter describes the principle aims of the research reported in this thesis. It introduces the overall concept of Production Data Analysis for discrete manufactured prismatic components within a contemporary metalworking SME, of which these aims form a significant part.

2.2 Research Objectives
The main aim of this research is to explore and investigate the requirements for a PDA computational framework which is capable of assuring both product and manufacturing process control within a contemporary metalworking SME. The proposed framework will endeavor to close the manufacturing feedback loop by addressing issues relating to the inspection activity and include:

i) Component measurement;
ii) Determination of geometric errors thorough comparative tolerance analysis;
iii) Determination of the most probable cause for the geometric errors;
iv) Recommend suggestions for corrective actions to eradicate the errors.

Although these essential elements of a closed loop manufacturing system have been recognised for the last few years the solutions proposed by both academic institutions and industrial vendors have concentrated on the first two elements only. This has produced fragmented systems that prove to be expensive, complicated and totally unsuitable for efficient application with the dynamic working practices experienced within a contemporary metalworking SME.

The main aim of the PDA framework will be addressed through the exploration of the complete spectrum of issues regarding a closed loop inspection system and is targeted at five major areas:
i) Machine and inspection planning of discrete component parts;  
ii) Production code generation;  
iii) Geometric error determination by comparative tolerance analysis of inspection results;  
iv) Production error diagnosis through the application of manufacturing data analysis;  
v) Information data resource integration.

Each research area is categorised through a number of main aims and objectives, which are outlined below.

2.3 Scope of Research
This research is written in the context of the trends and issues uncovered by an investigation into the next generation suite of tools to provide appropriate information technology (IT) support for the predominately human dominated activities exhibited within a contemporary metalworking SME. The research is heavily influenced by work carried out in the EPSRC GR/L27077 research project (Bell and Newman 1996) to investigate the most appropriate IT tools to improve the manufacturing performance of metalworking SMEs. The work expressed in this thesis is directed at closing the manufacturing information feedback loop that exists between the design/planning and manufacturing activities of a contemporary metalworking SME.

2.4 Production Data Analysis Framework
The Production Data Analysis framework aims to provide a seamless and integrated approach to production data code generation, for both manufacturing and inspection of discrete manufactured parts, and production data analysis to supply manufacturing performance feedback through the comparative tolerance and manufacturing data analyses of the inspection results.

2.4.1 Machine and inspection planning of discrete component parts
A novel concept for the amalgamation of relevant machine planning and inspection planning tools into an integrated system will be established. This concept, which will be built using an open CAM numerically controlled part programming package,
provides a versatile and powerful structure that is specifically suited to the interdisciplinary operational procedures conducted with a metalworking SME.

2.4.2 Production code generation
This work explores the requirements for the simultaneous generation of both numerically controlled part programs for CNC machining centres and post-process inspection programs for execution on direct computer controlled co-ordinate measuring machines with a single integrated system. This flexible and powerful production code generator enables the SME user to produce validated production programs transparently with little or no specialist inspection knowledge.

2.4.3 Geometric error determination by comparative tolerance analysis of inspection results
To research into a methodology to provide rapid and accurate geometric error determination of discrete manufactured components through the application of comparative tolerance analysis. This integrated approach will not require additional expertise or specialist training on behalf of the user without the need for an additional proprietary software application.

2.4.4 Production error diagnosis through the application of manufacturing data analysis
The work explores methods of production error diagnosis from the geometric errors determined from the comparative analysis of the inspection result against the components nominal specification in order to assist in the rapid recovery from manufacturing fluctuations that are not immediately apparent to the operator of the machine tool. The diagnostic tool is generic in nature to facilitate its application to different machine tools with the same basic configuration.

2.4.5 Information data resource integration
To investigate into the information modelling and database requirements of a contemporary metalworking SME. To identify the information specifications to support the functionality of the integrated Production Data Analysis framework. The framework utilises the existing information resources that exist within a contemporary metalworking SME. The Production Data Analysis framework aims to analyse these
existing information resources and augment them with the data produced by the combined production data analysis phases in the form of a quality information feedback model. This quality information feedback can subsequently be utilised to address production planning and control decisions.

2.4.6 Design of practical Experiments

In order to assess the validity of the production data analysis concepts and to highlight its effectiveness in closing the manufacturing information feedback loop will be achieved through a series of practical experiments. These experiments will involve the creation, machining and inspection and quality analysis of a feature-based 21/2 D prismatic type component and must concentrate on a variety of production error scenarios.
Chapter 3

PRODUCT INFORMATION MODELLING FOR MANUFACTURE

3.1 Introduction
This chapter introduces the concept of geometric and product modelling in the context of the data modelling requirements to support the automatic manufacturing and inspection procedures. This research review describes attempts to augment current solid modellers with additional component information to assist in downstream manufacturing activities. This chapter comprises of the following sections:

i) Geometric Modelling;
ii) Feature Based Modelling;
iii) Machine Operation Planning & Sequencing;
iv) Product Modelling;
v) Order Book Modelling;
vi) Manufacturing Resource Modelling;
vii) Information Modelling.

The author has included a brief outline of both geometric/feature-based modelling and machine operation planning/sequencing within the literature to provide the reader with a foundation for the remainder of the literature reported in this review.

3.2 Geometric Modelling
Geometric modelling techniques have evolved through the inability of computer aided design and drafting's (CADD) three view representation to provide the geometrical positional data required to support planning and manufacturing applications (Bedworth et al. 1991). This evolution has given birth to three types of geometric modelling techniques, namely wire frame, surface and solid modelling that can be utilised to describe the component's geometric representation.
3.2.1 Wire Frame Modelling

The wire frame modelling representation comprises of only points/vertices and lines/edges that possess true three-dimensionality. This representation does not contain information regarding surfaces and as a consequence of this wire frame models suffer from view ambiguity. Another disadvantage of the wire frame representation is that as there is an absence of connectivity information held with the model it is possible to create nonsense type objects (Bedworth et al 1991, Schutle et al 1992, Vosniakos 1998).

3.2.2 Surface Modelling

Surface modelling can be considered the natural progression from wire frame models with the addition of surface information, which may include planes, cylinders, spheres or sculptured surfaces. This type of representation is not capable of storing topology or connectivity information and therefore the model is unable to distinguish between the inside or outside of the part, moreover the constructed surface possesses an infinitesimally thin thickness. A ramification of these deficiencies is that calculation for volume and mass properties is not possible, whilst still allowing the user to produce nonsense type objects (Bedworth et al 1991).

3.2.3 Solid Modelling Techniques

Solid modelling defines the complete component geometry and topology (Case and Ago 1993) and contains mechanisms that remove the possibility of producing incomplete or nonsense type objects. Bedworth et al (1991) describes solid modelling techniques as consisting of two phases: solid model construction and solid model storage, together with six basic methods that can be employed to construct a solid model of a component:

i) Pure primitive instancing  iv) Sweeping
ii) Spatial occupancy enumeration  v) Constructive solid geometry
iii) Cell decomposition  vi) Boundary representation

As the latter two methods are the principle solid model construction representations upon which the others are based, the author will briefly discuss these methods in the rest of this section.
i) **Constructive solid modelling (CSG)** employs a technique of applying boolean operations on primitive solid objects to construct the component part. The boolean operators consist of union, difference and intersection and can be applied to the solid primitives: block, cylinder, sphere, cone and torus to produce complex solid model geometries. The solid geometry is captured in the form of a CSG tree that contains the locations and definitions of the primitive and their associated boolean operations. Leaf nodes of the tree represent the basic primitives employed in the solid model. The leaves are joined together by the operators. The combination of all the operators on the primitives constitute the complete solid representation of the component which forms the root of the CSG tree. The main disadvantage of the CSG method is that the CSG tree describes the geometry implicitly as the CSG tree must be evaluated every time the solid model representation is created or modified. This evaluation/re-evaluation can require a time period of between several minutes to several hour to complete, depending of the complexity and resolution of the solid model (Bedworth et al 1991, Requicha and Chan 1986, 1990, Case and Gao 1993, Shah and Rogers 1988).

ii) **A boundary representation (B-rep)** model is represented by its spatial boundary (Bedworth et al 1991) that consists of vertices, edges and faces. B-rep storage is larger than in a CSG representation and is stored in the form of a face-edge-vertex graph. It is because of the completeness of the graph-based model that it is often termed an evaluated model. A B-Rep model possess the capability of being associated with manufacturing information such as surface finish, material properties, dimensions and tolerances where the knowledge of the spatial boundary of the object is essential (Case and Gao 1993).

As both CSG and B-Rep solid model representations suffer from advantages and disadvantages, a number of researchers have experimented with a hybrid approach to solid modelling in order to exploit the advantages of both approaches (Gomes and Teixeira 1991). Although there has been considerable research undertaken into solid model representations (Requicha and Chan 1986, Gomes and Teixeira 1991, Shah and Rogers 1988, Case and Gao 1993, Allada and Anand 1995), B-Rep forms the basis of most commercial systems.
3.3 Feature Based Modelling

Many researchers have hailed feature-based technology as the key to the genuine integration of the design, planning and manufacturing activities (Case and Gao 1993). The potential benefits that can be exploited through the application of the feature-based approach can be summarised into three main domains (Shah et al 1988):

i) Designers can express easily and explicitly the design intent by manipulating features directly, eliminating tedious steps;

ii) Feature databases allow expert systems to perform tasks that may include heuristic reasoning, and manufacturing analysis; and

iii) Features can contain non-geometrical knowledge to facilitate NC programming, process planning and automatic finite element meshing thus providing the means of integration between design and manufacturing disciplines.

Although this area of research has reached maturity there is still no generic consensus on the definition of what represents a feature. This feature representation inconsistency has arisen as a consequence of two reasons:

i) The number of features that can be encountered is infinite;

ii) The requirements of feature definitions are application dependent.

These application dependent interpretations of features has led the CAM-I committee to specify unified definitions according to each manufacturing discipline (Shah et al 1988):

i) Design features - "Elements used in generating, analysing, or evaluating designs";

ii) Process planning or manufacturing features - "Shapes and technological attributes associated with manufacturing operations and tools";

iii) Geometric modelling features - "Groupings of geometrical or topological entities that need to be referenced together";

iv) Features in expert systems - "Objects formalised by a list of property slots, methods and inheritance hierarchy"; and

v) Database features - "Groups of associated or related elements".
3.3.1 Feature Taxonomies

As mentioned previously, the number of possible features is infinite. It was recognised by most of the researchers in this field that it could be conceivable to categorise features into groups or classes (Faux 1986, Butterfield et al 1986, Shah et al 1988, Shah 1991, Case and Geo 1993, Salomons et al 1993, Case 1994). Shah identified three major benefits from the utilisation of a hierarchical feature classification system, also known as a feature taxonomy (Shah et al 1988, Shah 1991):

ii) Features could be classified into families and the analysis would be designed to support those families as opposed to the supporting of individual feature configurations;

iii) The use of a feature classification could provide an aid to achieve some form of common terminology and standardisation; and

iv) Feature taxonomies can be employed to form the basis of product data exchange standards.

There have been a number of feature taxonomies identified by researchers for a variety of applications ranging from conventional machining (Faux 1986, Butterfield et al 1986, Gindy 1989) to aluminium extrusion, casting, injection moulding, sheet forging, sheet metal stamping, handling features and tool/die features (Cunningham and Dixon 1988). Such classification of form type features might include: passages, depressions, transitions, area features and deformations (Butterfield et al 1986).

There are three different approaches in which a feature database can be established in order to represent the component (Shah et al 1988, Shah 1991, Shah and Mäntylä 1995a, Allada and Anand 1995):

ii) Human assisted or interactive feature definition;

iii) Automatic feature recognition; and

iv) Design by features.

A comparison of the aforementioned feature modelling is outlined below (Shah 1991).
3.3.2 Human Assisted Feature Definition

With this method, the definition of features is achieved interactively by a human from an already complete geometric model (Shah et al 1991). The selected feature can then be augmented with attributes such as tolerances, surface finish etc. However, this form of feature definition is open to misinterpretation by the user. It must be noted that the interactive feature selection method is dependent upon the solid modelling scheme adopted. Human assisted definition is very easy to implement although it can be very time consuming if very large feature models are to be constructed.

3.3.3 Automatic Feature Recognition

Solid modellers store information in terms of low level entities such as vertices, edges and faces, in the case of a B-rep solid model, or CSG binary trees containing boolean operators. The process of automatic feature recognition attempts to find and extract application specific form features from a traditional solid model (Shah et al 1988). The recognised form features are stored in a separate database that forms the feature model (Case and Gao 1993, Brooks and Wolf 1994, Laakko and Mäntylä 1994a). Feature recognition can be categorised into five significant approaches (Lenau and Mu 1993):

ii) syntactic pattern recognition;
iii) state transition diagrams;
iv) decomposition methods;
v) CSG (set theoretic) approach; 
vi) graph-based approach.

All of the aforementioned techniques possess a number of limitations, firstly, these systems can only extract data that is contained within the solid model database. Therefore, any geometrical or non-geometrical data not contained in the initial CAD model cannot be extracted. Secondly, the initial CAD model is an interpretation of the product model formulated in the designer’s mind. The feature model is an interpretation of the CAD model. This double interpretation could lead to possible translational errors within the resultant feature model (Lenau and Mu 1993). Several other disadvantages are also encountered:
i) Recognition is redundant effort;
ii) Feature interactions are difficult to recognise.

One principle advantage of this method is that recognised features can be application specific, i.e. form, manufacturing or even inspection features (Shah et al 1988).

### 3.3.4 Design by Features

With this methodology the designer creates the model directly in the form of features and not the lower level entities previously discussed (Salomons et al 1993, Laakko and Mäntylä 1994a). Designers construct the feature model from the selection of functional features, which very often differ from application specific features. The approach is the only suitable method for simultaneous or concurrent engineering applications. A number of difficulties in the application of this approach that require addressing (Shah et al 1988):

i) the number of possible feature configurations is infinite;
ii) data management problems are challenging due to the complexity, variety and quantity of data to be managed and manipulated;
iii) since features are application specific the need for feature recognition by each application is still required.

### 3.4 Machine Operation Planning & Sequencing

Computer-aided process planning (CAPP) represents the link between the disjoint functional facets of traditional CAD and CAM facilities. Although the importance of process planning was not realised until the 1970s, it is widely recognised that the automation of the process planning activity reaps considerable benefit for the organisation. These benefits include the following (Bedworth et al 1991b):

i) Increased planner productivity;     v) Reduced scrap and rework;
ii) Increased equipment utilisation;   vi) Reduced shop labour;
iii) Reduced set-up costs;            vii) Reduced in work in progress;
iv) Reduced tooling requirements;     viii) Reduced material usage.
The effective integration of both CAD and CAM through the application of CAPP technologies and practices is only achievable by a deployment of feature-based techniques (see section 3.3) (Bedworth et al 1991b, Butterfield et al 1986, Shah et al 1988,1991). The domain of CAPP employs the component design representation which utilises information regarding manufacturing methods, processes, equipment and process capabilities to produce detailed process plans and associated documentation to support the manufacturing, accounting, purchasing and production control phases of the enterprise (Bedworth et al 1991b, Maropoulos 1995a, 1995b). The functionality of a comprehensive CAPP system would include the following elements (Bell and Young 1989, Zhang and Alting 1994):

i) Design input;   viii) Operation sequence selection;
ii) Stock material selection;  ix) Cutting tool selection;
iii) Process selection;   x) Cutting parameter selection;
iv) Machine route sequencing;   xi) Cost/time estimation;
v) Intermediate surface selection;   xii) Plan generation;
vii) Set-up identification;  xiii) Part-program generation;
vii) Fixture/holding determination;

Variant and generative approaches have emerged as the two main methods employed to automate the functions of the process planning activity (Shah et al 1991, Bedworth et al 1991, Eversheim and Schneewind 1993).

3.4.1 The Variational Approach to Process Planning is much the simpler of the two methods and relies heavily on human interaction and Group Technology (GT) principles (Shah et al 1991, Bedworth et al 1991c). The component’s GT code is used to identify a generic process plan from a standard process plan database for a particular part family (Shiko 1992, Laakko and Mäntylä 1994b). The selected generic plan is then interactively modified by the planner to the individual requirements of that particular component. If the component does not comply with any of the existing part families then the planner creates a new generic process plan through directed interaction with the computer interface. The variant CAPP approach assists the planner to remember similar standard process plans but is incapable of capturing the process planner expertise required to adapt the generic plan to the specific part instance plan (Shah et al 1991).
Examples of commercial variant process planning systems include: CUTPLAN by Metcut and CAPP by CAM-I (Bedworth et al 1991b).

3.4.2 The Generative Approach to Process Planning involves the automatic creation of process plans from engineering specifications expressed in both textual and graphical form (Shah et al 1991, Bedworth et al 1991c). This approach selects processes and their sequences via some predefined logic, which is based on manufacturing process rules. This generative procedure is achieved through computer-based algorithms that produces a unique process plan for a component without the requirement for interaction with the planner. Examples of experimental process planning systems that operate using the generative approach include (Maropoulos 1995a, 1995b): PART (van Houten et al 1990, Lenderink and Kals 1992), GENPLAN (Gindy et al 1993) HutCAPP (Mäntylä et al 1989), EXCAP (Tang and Davies 1990), BEPPS-ROT (Isik and Mileham 1992,).

The author recognises that the domain of CAPP is a vast research area in its own right, the remaining parts of this section will concentrate on the certain elemental functions that constitute a part of a comprehensive CAPP system. These elemental functions have direct bearing on the research reported in the latter part of the thesis and include: operation planning, cutting tool selection, NC part program generation.

3.4.3 Machine Operation Planning
The successful operation of a CAPP system depends heavily upon the integrity of the part definition data that is obtained through the CAD/CAPP interface (Brooks and Wolf 1994). This vital connection usually involves either the development of a human interactive or automatic feature recognition system to extract application specific manufacturing features from the functional/design features or solid model primitives that are expressed within the solid model database (see section 3.3.3).

Once the component's representation is expressed as manufacturing features, the features can be associated with manufacturing operations. XCUT (Brooks and Wolf 1994) develops a process plan through the construction of a feature access graph that represents the operations precedence in the form of a decision tree. This graph is created through the analysis of the individual feature access directions contained with the feature descriptions. This form feature representation is similar to that of the external access directions (EADs) identified and adopted by the Loughborough
University - Feature Based Design System (LUT-FBDS) (Gindy 1989, Gindy et al 1993, Case and Gao 1993, Case et al 1994). Isik and Mileham (1992) devised the BEPPS-ROT CAPP module for rotational components in which the feature ordering of an initial process plan is iteratively refined through rule based analysis. This feature precedence analysis is achieved via the consideration of machine tool, work holding, geometric and tooling constraints. The feature-driven process based design methodology reported by Schulte et al (1992) employs the concept of process plan fragments that are associated with each feature. This fragment approach to process plan generation is also adopted by Tolouei-Rad and Bidhendi (1995) that employs the use of two expert systems in the preparation of milling operation plans.

In selecting the sequence of operations, the strategies available for utilisation by the process planner are to minimise the number of machines, minimise the number of set-ups, and minimise the number of tool changes in each set-up that are required to produce a component. XCUT decomposes each feature in the feature graph into machinable volumes that constitute a single machining process (Brooks and Wolf 1994). Machinable volumes are also decomposed into machining cuts, the ordering of which is undertaken by the IDEEA expert system. Expert systems have also been utilised for the optimal ordering of operations within a process plan, Schulte et al (1992) uses a rule based strategy to order operations based on a feature taxonomy structure, Tolouei-Rad and Bidhendi's (1995) LISP based expert system uses the concept of an index of priority (IOP) to sequence the feature operations. Bell and Young's (1989) machine planner orders machining operations by the deployment of sequencing constraints, contained within the component's product model representation (see section 3.5) (Young and Bell 1992), that group operations by cut type, operation type whilst minimising the number of tool changes.

3.4.4 Cutting Tool Selection

Once the operation sequence is determined, the next phase is the selection of appropriate cutting tools required to produce the component. Brooks and Wolf (1994) analyse the geometry and tolerance requirement of each machining cut produced by XCUT to establish specific cutting tool parameters i.e. tool type, diameter, tool material etc. Possible tool candidates are selected from a current tool inventory and are assigned in the form of a cutting tool list to the associated machine cut. When all machining cuts for the component have been evaluated cuts that have intersecting tool lists are grouped
together to minimise tool changes and associated with processing attributes such as speeds, feeds and depth of cut. Young and Bell (1992) select tool and associated cutting data from manufacturing specific structures contained within a product model and are selected through the application of feature based manufacturing methods and constraints (Bell and Young 1989, Young 1991). Tolouei-Rad and Bidhendi (1995) have applied a second expert system constructed using the VP-Expert system shell to select cutting tools and processing parameters for the process plan generated through the application of a LISP based expert system.

3.4.5 Tool Path Generation

The generating of tool paths for rotational components is a relatively simple process given the boundary representation of the part, and involves the offsetting of the profile and generating both linear and circular tool paths using interpolation principles (Shah et al 1991).

The generation of tool paths for the milling of 2 ½ D components from a solid model representation of the component possesses the problem of calculating tool path offsets for curves, which becomes very complex when considering tolerances, fixturing methods and interacting geometric elements. Shah et al (1991) reviewed NC tool path generation techniques that can be applied to solid model representations and categorised them into four groups:

i) Cell decomposition
ii) Volume decomposition
iii) Sectioning or slicing
iv) AI or geometric reasoning

With the advent of the design by feature technique, tool path generation can be achieved directly through the application of feature based parametric macros that specify the feature’s tool path on feature creation (Camtek Ltd 1995). The information can then be augmented to the feature’s process plan fragment (Brooks and Wolf 1994) or processing methods constraints (Bell and Young 1989, Young 1991). The tool path generation for the complete component only involves the joining of the feature path elements together in the order prescribed in the process plan.
3.4.6 NC Part Program Generation

The process of NC part program generation involves the conversion of cutter location data for each manufacturing feature and the optimised process plan into CNC machine executable code. Both integrated machine operation planning systems reported by Schulte et al. (1992) and Vosniakos (1998) generate the NC part program in the form of the neutral file format COMPACT II which can be subsequently post processed into machine dependent NC code. Bell and Young's (1989) machine planner possesses the capability of producing NC code directly into the machine specific format designated by the process plan. Tolouei-Rad and Bidhendi (1995) on the other hand, produce a manufacturing data file from the planning activity, which forms the input to SmartCAM, a PC-based commercial CAM System that provides tool path simulation and NC code generation functionality.

3.5 Fundamentals of Simultaneous Engineering

Designers have traditionally expressed their design intent, which includes, geometry, tolerances, materials and machining requirements, in the form of manual paper oriented engineering drawings. These engineering drawings provided the standard interface, which linked the design process with the planning and manufacturing tasks. Over the past 15 years, the development of modern information technologies and practices coupled with the increasing demand for product development have forced a re-evaluation of the whole product definition concept. The introduction of contemporary computer aided engineering techniques over the past 35 years has resulted in the increased availability of a large variety of geometrical-based approaches. These automated design and manufacturing systems combined with the potential benefit of the implementation of information technologies have focused attention onto integration, as opposed to isolated automation (Krause et al. 1993). These product development practices, whilst striving for information integration of both design and manufacturing activities, have been heavily influenced by the need for:

i) Improved product quality;
ii) The reduction in product life-cycle costs; and
iii) The reduction of product development lead times (Molina et al. 1995).
The philosophy of Simultaneous Engineering (SE), or Concurrent Engineering (CE) as it is also termed, provides the vehicle to potentially improve the performance of product development practices (Molina et al. 1995). Winner et al. (1988) defined the CE philosophy to product development as (Molina et al. 1995, Borja 1997):

"A systematic approach to the integrated, concurrent design of products and their related processes, including manufacture support. This approach is intended to cause the developers, from the onset, to consider all elements of the product life-cycle from concept through to disposal, including quality, cost, schedule, and user requirements".

Molina et al. (1995) believe that the constant availability of relevant, reliable and consistent product and manufacturing information is the key to the computer systems which support SE and must be able to:

i) Capture and represent product, manufacturing process and resource information;
ii) Provide immediate access to information about previous product or process design and present design information without loss of intent or detail;
iii) Offer immediate access to information about manufacturability, reliability, maintainability, safety, performance, and other elements of the life-cycle;
iv) Allow access to the most current state of the product or process configuration description as it is being developed; and
v) Keep data to be shared by team members in accessible databases.

It has been recognised through a number of experimental environments, i.e. IMPPACT (Meier 1990) and MOSES (Corrigall et al. 1992, Molina et al. 1995), that the minimum information requirements to support SE are primarily concerned with: product modelling, and manufacturing modelling (see section 3.7).

The remaining sections of this chapter shall concentrate on the modelling techniques employed to capture the product and manufacturing oriented information required for the design and production of mechanical components.

3.5.1 Product Modelling Concept
Mäntylä (1989) interprets product modelling in the context of integrated CAE techniques as:
"The activities relating to representing and utilising information related to products, their design and manufacturing processes and their production management".

The output of the product modelling activity is the product data model. This product data model describes the generic structure of a product model, whilst the product model is a model instantiated with actual product data that has been captured within the structure of the product data model (Shaw et al 1989, McKay 1991). The accepted definition of a product model was proposed by Krause et al (1993a):

"A product model, is the logical accumulation of all relevant information concerning a given product during the product life-cycle. They store the information in the form of digital product model data, and are equipped with access and manipulation algorithms".

The product life-cycle refers to the period of time from the inception of the design to the disposal or recycling of a product. The types of information flows within the life-cycle of a product from an industrial perspective is portrayed in figure 3.1 (de Pennington 1990). This diagram illustrates the processing stages in which a manufactured product may experience during its life-cycle. It can be noted that there are a number of positions in the product's life-cycle that can influence the development of subsequent products through the effective use of information feedback as indicated at the inspection and test stages of figure 3.1.

![Figure 3.1. Product Life-Cycle (de Pennington 1991)](image-url)
The type of information captured by the product data model range encompasses all the necessary product geometry, topology, functional and technological features, dimensions, tolerances, surface finish, material properties etc. (Domazet and Manic 1990). The product model is populated, utilised and subsequently re-populated through the communication and execution of the various design and manufacture activities that form a part of the integrated CAE environment. These computer-aided systems include computer-aided design, (CAD), computer-aided process planning (CAPP), NC part programming (CAM) and computer-aided inspection (CAI) etc.

Product models can be application specific, in the sense that the information contained within them only supports a number of applications, such as planning, manufacturing and inspection (Atkins 1989, Jasthi et al 1993). Jasthi et al identified the requirements for an application specific product model to assist in the computer-aided process planning (CAPP) task. Their approach focused on three types of data:

i) **geometrical data** - description of geometry or shape;

ii) **technological data** - describes tolerances and surface finish characteristics;

iii) **global data** - describes quantity required, design number, part name, department and other task dependent details.

On the other hand, integrated product models can be created to cover an enterprise wide perspective. In the light of this, due to the bewildering amount of information that a product model can possess, the model can be decomposed into a number of sub-models that can be clustered together to facilitate model maintenance (Krause et al 1993a, Nnaji et al 1993, Reimann et al 1993, 1994, Yan et al 1997). Krause et al has categorised these sub-models into structure-oriented product models, geometry oriented product models, feature-oriented product models, knowledge-based product models and integrated product models.

i) **Structure-oriented product models** capture the products structure in the form of the product’s breakdown provides the kernel of this type of product model. This structural information can include: bill-of-material structures, classification structures, product variant structures, order processing information, access functions and computer network addresses. An example of a structure-oriented

24
product model employed for the design and production of ships and offshore structures, such as oil rigs, is the AUTOKON system (Krause et al. 1993a).

ii) **Geometry-oriented product models** are primarily concerned with the capture of the geometric representation of a specific product. These representations can consist of wire frame, surface, solid or hybrid type models (see section 3.2). Early examples include: PADL and GEOMOD (Spur et al. 1979).

iii) **Feature-oriented product models** provide an extension to that of geometric models by providing the ability to represent patterns of geometric entities in the form of features (see section 3.3).

iv) **Knowledge-based product models** are categorised by the application of artificial intelligence techniques, i.e. object-oriented programming, rule-based reasoning, constraint and truth maintenance systems. This type of product model is capable of capturing human expertise, experience relating to products, processes and factory environments. Knowledge-based models are employed to enhance the capabilities of information support during the product modelling process. The Intelligent Design Environment for Engineering Applications (IDEEA) utilised within the XCUT expert planning system (Brooks and Wolf 1994) provides an example of a knowledge-based product model.

v) **Integrated product models** are those types of models that comply with the definition of a product model quoted earlier (Krause et al. 1993a). They encompass the abilities of structure-oriented, geometry-oriented, feature-oriented and knowledge-based product models. Integrated product models are therefore capable of capturing all relevant information concerning a given product during the product life-cycle.

### 3.5.2 Integrated Product Modelling Environments

One of the most significant contributions to the realisation of an integrated product modelling environment was the application oriented approach adopted by ESPRIT project 2165 known as the Integrated Modelling of Product and Processes using Advanced Computer Technologies (IMPPACT) (Meier 1990, Bjarke and Myklebust 1992, Gielingh and Suhm 1993a, Krause et al. 1993a). The 3 year project involved 14 partners distributed through Europe and Scandinavia and dealt with the creation of product, factory and process models for discrete part manufacturing. The objectives of the IMPPACT project was (Gielingh and Suhm 1993a):
"To develop and demonstrate a new generation of integrated modelling systems for product design and process planning including machine control data generation. A reference model as a general approach for deriving specific software strategies builds the framework. A flexible, adaptable architecture will be used, open for the integration of future software components. The integrated product data model contains all information about the product and it's production processes. Limitations of integration will be overcome by a conceptual approach of using features in all stages of the manufacturing process."

The results of the IMPPACT initiative have heavily influenced the development of the ISO10303 Standard for the Transfer and Exchange of Product Model Data (STEP) (see section 3.5.3.4) and was summarised by Krause et al (1993a) as providing:

i) An information modelling methodology for product modelling;
ii) Generic and specific reference concepts as applied to discrete part manufacture;
iii) System components for integrated product and process modelling;
iv) An EXPRESS based object-oriented database management system for the integration of distributed database systems.

The approach identified through the IMPPACT project was applied to two areas of discrete part manufacture namely: sheet metal part manufacturing for aircraft spares and complex shape component manufacturing for ship propellers.

A model-based approach was taken by Kimura (1991,1993b), at the Precision Machinery Engineering Department, The University of Tokyo, for developing a software framework for Product Realisation. The Product Realisation approach encompasses the Simultaneous Engineering philosophy. This model-based framework, enabled models to be classified into either computer executable, i.e. object and activity models, or abstract and logical models. Object and activity models are categorised by their physical constraints. Kimura identifies three object type models:

i) **Product Model**: represents every artefact with its related physical realisibility constraints and evolving definition.

ii) **Manufacturing Resource Model**: represents the virtual factory. It consists of models of existing resources within the factory.
iii) **Physical Model:** represents the physical processes and is employed to investigate product functionality and manufacturability.

Activity models represent the processing activities of the system and are classified by Kimura (1993b) as:

i) **Integration Model:** is a meta-model that defines the overall structure of the system and serves as the foundations for the other models.

ii) **Design Process Models:** represents all manufacturing engineering activities.

iii) **Production Management Model:** is used for managing manufacturing resources.

Kimura has recognised that for an efficient manufacturing software implementation the systems need to be of modular design, adaptable and allow human interaction through the application of carefully constructed user interfaces.

A computer aided engineering system to support simultaneous engineering called the Model Oriented Simultaneous Engineering System (MOSES) research project provides another example of an integrated approach to product modelling (Ellis et al 1995). This research project was conducted as a joint project by Loughborough University and Leeds University identified the importance of the role a product model (Shaw et al 1989, McKay 1991, McKay et al 1996,1997) and a manufacturing model (Molina et al 1994, Molina 1995, Al-Ashaab and Young 1994, Molina and Bell 1999) in CAE systems of the future. The product model captures all the data related to the product’s life-cycle whilst the manufacturing model contains all the information regarding the manufacturing facility in the form of process capabilities, manufacturing resources and strategies. The architecture of MOSES is depicted in Figure 3.2 which illustrates the product and manufacturing data models link through an integrating environment to a number of application environments (Ellis et al 1995).

The Engineering Moderator ensures the evolving product designs consider the different life-cycle requirements represented by the application environments whilst resolving any conflict that may exist. All product related data is stored within the product model as the design evolves. If an application is initiated it utilises information contained within the product model and subsequently augments the product model with data generated through the execution of the application. The manufacturing process...
information required during the execution of an application is sourced from the manufacturing model (see section 3.7). Although MOSES is capable of supporting any number of application environments only the design for manufacture (DFM) has been implemented.


![Product Model](Product Model) ![Manufacturing Model](Manufacturing Model) ![Engineering Moderator](Engineering Moderator) ![Integration Environment](Integration Environment)

**Figure 3.2.** The Structure of the Model Oriented Simultaneous Engineering System (MOSES) (Ellis et al 1995)

### 3.5.3 Data Transfer Standard Formats

Computer databases are now replacing paper oriented engineering drawings as the main means of defining product geometry and non-geometry for all phases of product design and manufacturing. The fundamental incompatibility that exists between the different information representations of a multitude of CAD/CAM systems is severely hampering the exchangeability of modelling data. The transferring of data between dissimilar CAD/CAM systems must include the complete product definition held within its database. This definition has been acknowledged to contain four types of modelling data (Zeid 1991):
i) Shape data - geometrical and topological information, part or form feature, entities such as font, colour etc. are contained within this type of data;

ii) Non-shape data - this type includes graphical data such as shaded images and the resolution of storing the database numerical values;

iii) Design data - this type is concerned with the information generated from analysis of the geometric models such as mass property and finite mesh data;

iv) Manufacturing data - consisting of tooling, NC tool paths, tolerancing, process planning tooling details and bill of materials.

Neutral data formats, that are designed to communicate product definition data between dissimilar CAD/CAM systems, must incorporate all the aforementioned categories of data otherwise the scope of such a data format will be severely restricted.

3.5.3.1 Initial Graphics Exchange Specification (IGES)

In September 1979 an Air Force integrated computer aided manufacturing (ICAM) program, which consisted of representatives of both government and industry, was initiated to develop a method of data exchange. A technical committee, comprising the Boeing Company, General Electric Company and the then called National Bureau of Standards was established and assigned to the task. The result of their efforts was the publication of the Initial Graphics Exchange Specification (IGES) in January 1980 (Zeid 1991). The version 1.0 specification was adopted as the ANSI Y14.26M standard by the American National Standard Institute in September 1981 (Owen and Bloor 1987). This standard has been upgraded on a number of occasions (National Bureau of Standards 1988, Reed et al 1990) and currently is at version 5.3, which was released in 1996 (Anon. 1999). Although IGES was originally conceived to facilitate the exchange of geometrical and non-geometrical data between CAD/CAM systems of the types encountered in the 1970s and 1980s, IGES now can support the exchange product data models in the form of wire frame surface or solid representations. Applications supported by IGES include traditional engineering drawings and models for analysis and/or various manufacturing functions (Reed et al 1990).

The initial efforts of IGES provided the catalyst that stimulated interest into the defining of better exchange standards for representing the four fundamental types of product modelling data (Owen and Bloor 1987). Examples of which are the Standard d'Echange et Transfert (SET 1984) developed by the French company Aerospatiale and
released in 1983 supported the similar range of entities as IGES but in a free format unlike IGES version 1.0. This afforded the SET specification enhanced performance and flexibility over IGES. The German TAP and VDA-FS standards (VDA-FS 1984) developed, as an alternative to IGES, by the automotive industry for the exchange of free-form surface data and was found to be limited in the precision of the surface definitions (Corrigall 1990). With the widespread acceptance of solid modeller principles the Computer Aided Manufacturing - International (CAM-I) organisation funded a project to develop a solid model exchange standard known as the experimental boundary file specification (XBF) (Owen and Bloor 1987). Published in 1981 and revised in 1982, XBF files were capable of supporting both CSG and B-rep model representations. The efforts of the XBF and IGES initiatives were closely related and as a consequence of this XBF was merged with IGES to form the IGES Experimental Solids Proposal (ESP) (Owen and Bloor 1987).

The file format defined by IGES describes the product definition as a file of entities in an application independent format, to and from which the native representation of a specific CAD/CAM system can be mapped (National Bureau Standards 1988, Standards Association of Australia 1989).

3.5.3.2 The Product Definition Data Interface (PDDI)
The product definition data interface (PDDI) was developed by the US Airforce ICAM project and was awarded to the McDonnell Aircraft company in 1984 for the determination of long-range manufacturing needs and for a prototype demonstration of a product definition data interface to accomplish those needs (Shah, J et al 1988, Atkins et al 1989). The PDDI project’s initial aim was the eventual replacement of the engineering drawing and expanded the IGES concept through the incorporation of additional manufacturing information. The system was intended to serve as the information interface between engineering and all manufacturing activities, such as process planning, NC programming, NC verification, quality assurance, tool design, robotics and others (Zeid 1991). The data elements that were supported by the PDDI system were categorised into five areas:

i) geometry; iv) form features; and
ii) topology; v) control information;
iii) tolerances;
The significance of the PDDI project was that it was not restricted to geometry and topological entities as the identification of form features was considered as an important part of the product data representation. Although the PDDI system could cater for discrete mechanical components of either sheet metal, turned, composite and machined configurations it could not deal with component assemblies.

3.5.3.3 Product Data Exchange Standard (PDES)
Product Data Exchange Standard (PDES) was a project initiated in 1984 by the IGES organisation in order to establish a mechanism for complete product model data exchange (Shah et al 1988). PDES was directly influenced by the plethora of data exchange contributions from IGES, XBF, IGES ESP, and PDDI. PDES was designed to facilitate the data exchange within a number of applications, including architecture, engineering, construction, mechanical features, finite element modelling, and printed circuit board manufacture. PDES also addresses technological support such as topology, geometry of both CSG and B-rep solid models, tolerancing, presentation and administrative data.

PDES adopts a rigorous methodology that is employed in the design of databases in the construction of its information models which consists of a three layer architecture:

i) **Application or external layer** - Individual information models for each application are modelled independently, from the user's point of view, at this level;

ii) **Logical layer or conceptual model** - The individual application models are integrated into a single information model of the total system. The information at this level refers generically and not specifically to all the applications in the external layer. Normalisation is performed at this level in order to remove redundancy;

iii) **Physical or internal layer** - Here the conceptual model is converted into a specific file format. This layer contains the specification of sections, records, field, sequencing and associated formats for the exchange file.
Research into the exchange of product definition data was not only confined to the United States, in 1984 a five year ESPRIT research project, led by Germany, focused on European developments in data exchange standards (Zeid 1991).

3.5.3.4 ISO Standardisation and the Standard for the Transfer and Exchange of Product Model Data (STEP)

With the now bewildering array of emerging data exchange standards, with no common format, it was recognised that there was an essential need for some form of standardisation. Therefore, a subcommittee, SC4, was formed in 1984 within the ISO Technical Committee TC184 (Industrial Automation Systems) to centralise and manage all data exchange development. The subcommittee agreed that there was a call for a single global standard for the exchange of product definition data. This global standard, which is identical to PDES, was termed the Standard for the Transfer and Exchange of Product Model Data (STEP) or ISO10303.

A basic introduction into STEP in terms relevant to the needs of decision makers who require to understand the impact of product data technologies on today's business environment is discussed by both Owen (1993) and Fowler (1995). The main goals of the STEP initiative can be summarised as a set of nine objectives (Owen 1993):

i) **Completeness** - STEP should allow a complete representation of a product, for both exchange and archiving purposes;

ii) **Extensibility** - STEP must provide a framework into which extensions of domains can be created;

iii) **Testability of additions** - before any additions are to be incorporated in the standard it should be subjected to a review and, if possible, undergo further testing by being implemented;

iv) **Efficiency** - STEP should be efficient both in terms of file size and the computer resources needed for processing;

v) **Compatibility with other standards** - STEP should be compatible with other standards in order to ease migration from existing standards. It should employ facilities for other standards where applicable;

vi) **Minimal redundancy** - there should be only one way of representing a particular concept;
vii) *Computing environment independence* - STEP should be independent of particular hardware and software;

viii) *Logical classification of data standards* - STEP should define subsets for implementations as it would clearly be a substantial standard; and

ix) *Implementation validation* - a framework of conformance testing should be part of the standard.

STEP (ISO 10303) is a substantial international standard for the computer-interpretable exchange of product data and is intended to provide a mechanism for describing product data throughout the life-cycle of a product, independent of any particular system (NIST 1999). STEP is not only directed at neutral file exchange, but also provides a basis for implementing and sharing product databases and archiving.

The STEP standard consists of an extensive number of parts that are grouped together to reflect the structure of the standard as described in ISO10303-1 (NIST 1999, ISO/DIS 10303-214 1999). These groupings can be applied to diverse applications ranging from draughting, electrotechnical design and installation, building structures and ship building and comprise of the following:

i) Parts 11 to 13 specify the description methods;

ii) Parts 21 to 26 specify the implementation methods;

iii) Parts 31 to 35 specify the conformance testing methodology and framework;

iv) Parts 41 to 49 specify the integrated generic resources;

v) Parts 101 to 106 specify the integrated application resources;

vi) Parts 201 to 233 specify the application protocols;

vii) Parts 301 to 332 specify the abstract test suites; and

viii) Parts 501 to 518 specify the application interpreted constructs.

Unlike the other standardisation activities, STEP has a forward looking and not retrospective viewpoint as a number of experimental product modelling systems have been developed based upon the STEP standard and commercial system interpreters (Nakamura *et al* 1993, Huang *et al* 1994, Yan *et al* 1997, Demartini *et al* 1998, Pratt 1998, Dutta *et al* 1998).
3.5.3.5 Dimensional Measuring Interface Specification (DMIS)

The Dimensional Measuring Interface Standard (DMIS) is a neutral data exchange format specification for the bi-directional communication of inspection data between CAD/CAM computer systems and computerised inspection equipment (Aubin 1987, ANSI/CAM-I 1990, International Metrology Systems 1995, Sivayoganathan et al 1995). DMIS version 1.0 developed by the Illinois Institute of Technology Research and funded by CAM-I was offered for public release in April 1986. Subsequent upgrades, versions 2.0 and 2.1, were developed by Pratt & Whitney and TechTran Inc. respectively. DMIS version 2.1 was adopted by the ANSI as the American National Standard ANSI/CAM-I 101 in 1990.

The DMIS vocabulary is of similar construction to the Automatically Programmed Tool (APT) NC programming language and specifies a neutral format for both inspection programs and results data. DMIS has been designed to be human readable and writable, thus allowing programs to be constructed without the aid of computer technology. DMIS incorporates the statements and commands necessary to define the part geometry and drive both CMMs and machine vision inspection systems for the dimensional inspection of mechanical parts. There are two basic types of DMIS statements: (i) process-oriented commands; and (ii) geometry-oriented definitions.

Process-oriented commands comprise motion statements, machine parameter statements, and other inspection process related statements. Geometry-oriented definitions are used to describe geometry, tolerances, co-ordinate systems, and other data associated with the CAD database (ANSI/CAM-I 1990, International Metrology Systems 1995). The DMIS specification defines only the vocabulary transmitted by the American Standard Code for Information Interchange (ASCII) files. The method of transmission, storage, and management of the ASCII files is dependent upon the user.

The development of DMIS has been closely monitored by the committee responsible for the development and maintenance of the IGES (ANSI Y14.26M) specification. Therefore, DMIS was created to evolve and maintain compatibility with the future developments of the IGES specification and is currently at version 3.0 which was approved as an American National Standard in February 1996 (CAM-I 1999).

3.5.4 Critique - Simultaneous Engineering Approaches

The simplest form of data exchange for integration involves the creation of specific interfaces that allow information to pass from one application to another. This approach
has the advantages of minimal disruption to the application themselves whilst allowing the system to evolve with the creation of additional interfaces when new applications are introduced. However, the disadvantages associated with this approach increase as the number of applications grows, and are identified as:

i) Engineering data management and control is difficult;
ii) Data duplication is experienced within applications. From the perspective of each individual application their version of data is the master;
iii) Interfaces between certain applications may prove to be impossible to write resulting no integration and no data exchange;
iv) The number of interfaces dramatically increases as the number of applications grow making them difficult to maintain.

Information exchange through the application of neutral exchange formats offers enormous benefits over direct interfacing for device independent communication of data between dissimilar computer aided engineering facilities, however there are also a number of problems that can be encountered (Owen 1993):

i) Neutral data formats require large amounts of development time as they are established through voluntary efforts by a democratic process, e.g. STEP;
ii) These formats usually form retrospective standards that are out of date at the time of release;
iii) Neutral data formats suffer from limited coverage;
iv) Conforming to a neutral data format does not guarantee bi-directional data exchange. If two applications employ the same neutral data format, e.g. DMIS, for data exchange but support the standard to differing degrees of coverage, data exchange will be possible from the application with the lower coverage to the application with the greater coverage, however, this may not be possible in the opposite direction;
v) They require a pre- and post-processor facility to accomplish data exchange as opposed to just a post-processor as is the case for direct translation.

Although the application of the integrated product model approach of a single, unique central source of information shared by various applications negates data
duplication problems and eases engineering data management and control of the information, there are a number of problems that stem from their complexity that require addressing:

i) The generic form of a production data model is still not available although the application of the STEP standard is helping alleviate this problem;

ii) Methodologies and mechanisms for reducing the complexity of product data models is still the subject of much research. The complexity of these models will continue to increase as the number of application environments increase;

iii) The way in which the structure of product data models is determined is not always obvious, usually iterative, and time consuming.

3.6 Order Model Concept

Since the application of the product modelling approach to integration is very complex, time consuming and expensive to adopt, it tends to be applied to large companies that are very much deterministic in nature who manufacture static product ranges in large batch quantities. This approach proves to be an inappropriate solution for the Small Manufacturing Enterprise (SME) operating at end of the supply chain. This sector of companies produce a diverse product range in small batch quantities whilst operating within volatile customer driven markets. Hvam (1995) has identified the need for an adapted product model environment to support the SME. Although the solution reported was designed and implemented, within a medium sized Danish SME comprising of 200 employees, to model a specialist type component consisting of a large number of part variants, it does not address the requirements for smaller SMEs.

Nicholson (1985) and Westbrook (1993) introduced the notion of using a data structure that captures order-oriented information for the purpose of priority management. The priority management approach is the expression of a preference, to specific order or order groupings (i.e. supplies, production or customer orders) in response to current pressures on operational productivity and customer service (Toh et al 1998). It is because of the suitability of this approach to cope with the volatility exerted by customer markets on the SME that has influenced the inclusion of an order-based information as part of Toh’s SME enterprise model (Toh 1997). The SME reference model which form part of the EPSRC nationally funded research programme (GR/L27077) into identifying appropriate “IT tools to improve the manufacturing
performance of metalworking SMEs” (Bell and Newman 1996) and consists of three sub-models: an order sub-model, a manufacturing sub-model and an organisation and behaviour sub-model. This research programme is discussed further in chapter 6 of this thesis.

The order sub-model in its basic form defines a core information structure relating to the progress, status and location of customer orders in the form of order and job class structures (Toh 1997, Toh et al 1998). The information captured by the order sub-model caters for different users throughout the factory from process planning and manufacture. Although the order sub-model is geared to support an SME without a product design function it has been recognised that it represents a sub-set of information captured by a component product model.

3.6.1 Critique – Order Modelling
The order sub-model offers a solution to the complexity issues of product modelling when applied to an SME, and provides the potential to be extended to include information to support manufacturing activities such as machine and inspection planning and production code generation (Bagshaw and Newman 1998, 1999). This gives the order sub-model the ability to evolve over a time period to provide similar coverage and support to that promised by a component product model whilst being of manageable complexity thus facilitating model creation and maintainability.

3.7 Manufacturing Resource Modelling
Numerous researchers have recognised that many applications require not only information regarding the product but also a consistent source of manufacturing information. The data model that captures manufacturing information is referred to as a factory model (Bjørke and Myklebust 1992, Gielingh and Suhm 1993a), manufacturing resource model (Kimura 1993b) or a manufacturing model (Molina 1995, 1999).

Pioneering work into the modelling of manufacturing information was conducted as part of the IMPACT project (Bjørke and Myklebust 1992, Gielingh and Suhm 1993a). IMPACT defines a factory model that represents the real production systems components, known as production mechanisms, and the physical and logical relation between them (Gielingh and Suhm 1993a). These production mechanisms cover the technological and human resources referenced by the process and operation planning activities and other existing means of production. The factory model
structures the resource information for machine tools, cutting tools, tool assemblies, transportation equipment etc., as well as administrative information regarding operator attributes, performance indicators and maintenance information at shop floor and work centre level. A Process Model consists of the information that describes the production activities that can be performed utilising manufacturing resources such as manufacturing processes, operations and tool paths. The factory, process and the product model (see section 3.5.2) all possess shared information that provides the vehicle for application integration.

Kimura's (1991) Manufacturing Resources Model attempts to represent the whole factory in the form of information relating to machine tools, cutting tools and assemblies, fixtures, jigs, control devises, communication equipment, materials, buildings and human resources etc.

The Manufacturing Model of the MOSES project (Ellis et al 1995, Molina et al 1994, Molina 1995b)(see section 3.5.2) forms the second key data model of the simultaneous engineering environment. The manufacturing model describes and captures the information regarding the manufacturing situation of the company in terms of the manufacturing facility and capabilities at factory, shop, cell and station levels of abstraction. The MOSES manufacturing model identifies three basic elements that are employed to define any manufacturing environment: resources, processes and strategies. These elements, relations and interaction between them are considered fundamental to describing any type of manufacturing enterprise (Ellis et al 1995).

Manufacturing resources consist of all the physical elements within a facility that enables product manufacture. Manufacturing processes are the activities that are carried out within the facility in order to produce a product and manufacturing strategies represent constraints and decisions made on the use and organisation of both the manufacturing resources and processes (Molina 1995, 1999).

3.7.1 Critique – Manufacturing Modelling

Product models by their very nature are dynamic models as they evolve as the product progresses through its life-cycle, from start to finish. Manufacturing models, on the other hand tend to be static in nature. Once created to represent the complete manufacturing facility of a company they remain static and only require modification when new manufacturing resources, processes and strategies are introduced into the company. It has been recognised by numerous research initiatives, e.g. IMPPACT and
MOSES, that to support the design to manufacture phases of product realisation requires not only product information captured by an unambiguous product model, but also consistent manufacturing information contained within a manufacturing model. However, to put this in context, in a company wide integrated modelling environment there may be a number of evolving product models within the company, one for each product produced, there will be only one manufacturing model to represent the manufacturing facility. Although STEP attempts to standardise product models and their data exchange there has been no attempt at present to standardise how manufacturing information should be structured. This has resulted in a great diversity of the approaches adopted by active research groups for manufacturing modelling.

The reliance on both the information models to support efficient integration increases the complexity issues of the information system which are best suited to large companies that are deterministic in nature, and reinforces its unsuitability for adoption within a metalworking SME.

### 3.8 Information Modelling Techniques

Information modelling, also known as data or semantic modelling, is the process which results in the generation and a data or semantic model. Information modelling is concerned with the structure of data and as a consequence of this information models are application and processing independent. These semantic models form the basis for the design of application independent databases and interfaces (Toh 1997). There are many tools available to assist the modeller to define the data, its structure and relationships. These include both graphical and textual representations.

Coad and Yourden (1990) proposed an object oriented method for the structured analysis approach to software engineering that includes textual and graphical representations of data and relationships between data. These tools include: event lists, context diagrams, data dictionaries, data flow diagrams (DFDs), state transition diagrams (STDs) and structure charts.

IDEFO is an abbreviation for ICAM Definition level 0 (Colquhoun et al 1993, Meta Software Corporation 1995) and is based on the Structured Analysis and Design Technique (SADT) and is used for functional/activity analysis of the systems. This approach is a top down graphical method for identifying the activities of the system. The system is initially envisaged as one single global activity. The system is then decomposed into sub-activities that are required to achieve the global activity. Sub-
activities are in turn decomposed into a lower level of abstraction. This process continues until the system is described in sufficient detail. IDEF0 has been employed for functional analysis of information modelling systems: MOSES (Molina et al 1994), IMPPACT (Bjørke and Myklebust 1992, Gielingh and Suhm 1993b).

IDEF1x is a modelling language that is used to produce a graphical information model, which represents the structure and semantics of information within a system (Bruce 1992, Meta Software Corporation 1995, IDEF 1999). IDEF1x models data in a standard, consistent and predictable manner in order to manage it as a resource. IDEF1x’s primarily application is with relational type database systems.

Another popular method used for data modelling and was employed in the IMPPACT project (Gielingh and Suhm 1993c) was developed by Nijssen in 1967 and called NIAM. NIAM models consist of four basic elements: object types, fact and fact types, subtypes of objects and constraints.

EXPRESS is the formal data specification language specified by the STEP initiative. The use of this type of formal language enables a consistent representation of the information held within data models (Gielingh and Suhm 1993c, Rahimifard and Newman 1996). EXPRESS-G is a graphical representation of the entity and attribute relationships. As EXPRESS data models can be interpreted by computer it can be tested, validated and readily converted into a more convenient format according to both user and system requirements. The formal language has been utilised for both the IMPPACT (Bjørke and Myklebust 1992, Gielingh and Suhm 1993d) and MOSES (Molina et al 1994) projects.

The Booch methodology is applied to object oriented systems and is concerned with the decomposition of systems, the activities that occur within the objects and the interaction between objects, through the application of class and inheritance diagrams (Booch 1994). The methodology involves three phases (Booch 1994, Molina et al 1994, Toh et al 1998):

i) Conceptualisation endeavours to establish the requirements for the system in the form of high level statements describing the system’s purpose and scope;

ii) Analysis determines the logical structure of the system; and

iii) Design establishes the physical structure of the system based on the logical structure which proceeds to a working prototype.
The Unified Modelling Language (UML) (Anon 1999) has evolved from the Booch methodology and is one specific language that provides system developers working on object analysis and design for specifying, visualising, constructing and documenting the artifacts of a software system as well as being able the support business modelling. The UML approach represents a collection of best practices that have proven successful in modelling of large and complex systems in terms of the views available of a model, the UML defines the following graphical aid tools:

i) use case diagrams;
ii) class diagram (Zhao et al 1999);
iii) behaviour diagrams: statechart diagram, activity diagram;
   interaction diagrams: sequence diagrams;
   collaboration diagrams;
iv) implementation diagrams: component diagrams, deployment diagrams.

Rahimifard and Newman (1996) identified a methodology for describing the stages involved in a data modelling which utilises standard modelling tools that can be employed at each stage of the modelling process in order to guide the developer through the development of a data model from start to finish. The stages of the modelling process can be summarised as:

i) Identification of data requirements for the system using input/output diagrams;
ii) Construction of a data index;
iii) Modelling of information flows using IDEF0 diagrams;
iv) Specification of the functionality of the system using YOURDON techniques;
v) Specification of the entity relationships via an EXPRESS-G diagrams;
vi) Development of the data model through the application of the EXPRESS data modelling language.

The Structured System Analysis and Design Methodology (SSADM) (Ashworth and Gooland 1990) is a structured method for producing logical and physical design specifications for computer system applications. It provides a step by step evolution from an existing system to the desired system. SSADM consists of six stages (Ashworth and Goodland 1990, Hares 1990): (i) analysis of system operations and current
problems, (ii) specification of requirements, (iii) selection of technical options, (iv) data design, (v) process design, and (vi) physical design.

Stages i), ii), iii) and vi) are executed in sequence, whereas stages iv) and v) occur concurrently. SSADM employs ten analysis techniques: logical data structuring technique (LDST), data flow diagrams (DFDs), relational data analysis (RDA), composite logical data design (CLDD), entity life histories (ELH), process outlines (PO), logical dialogue design (LDD), first cut design (DD), physical design control (PDC) and program specifications (PS) (Hares 1990).

3.8.1 Critique – Information Modelling Techniques

It can be recognised there are a plethora of modelling tools that can be employed to aid in the modelling of information. These tools provide a tool kit from which the systems developer can choose from in order to express the content and structure of data models. Although these are modelling methodologies, such Yourdon, Booch, SSADM, Rahimifard and Newman guide the developer through the system analysis, requirements specification, enabling technology selection, data and process design and physical system design phases of the modelling process, the tools used at each phase will depend upon the application. There is no consensus as to the preferred combination of modelling tools to apply to a particular application. An exemplar of this is the methodologies employed for the data modelling of products and processes employed by the IMPPACT and MOSES modelling environments. IMPPACT employed tools such as IDEF0, NIAM and EXPRESS, whereas MOSES utilises IDEF0, BOOCH and EXPRESS for the modelling process.

The Unified Modelling Language is currently the most popular vehicle for standardisation of object design and analysis that can be applied to the representation of products, processes and enterprise models.
Chapter 4

REVIEW OF CONTEMPORARY AUTOMATED INSPECTION RESEARCH

4.1 Introduction
This chapter describes the contributions of various academic and industrial institutions into contemporary automated inspection research. It introduces the novel approaches employed by academia to solve the complex activities that are required to plan, execute and analyse the component inspection data on co-ordinate measuring machines (CMMs). This chapter consists of the following sections:

i) Tolerancing Methods and Standards;
ii) Inspection Planning for CMMs;
iii) Inspection Code Generation;
iv) Inspection Result Evaluation;

The author recognises that a significant contribution to the advancement of automated inspection systems has been through the adoption of visual inspection techniques, which forms a vast topic of research in its own right (Chang *et al* 1988, Marshall and Martin 1992, Ventura and Chen 1994, Newman and Jain 1995, Smith *et al* 1995, Noble 1995). However, the research reported in the rest of this thesis will concentrate on the automated inspection of manufactured components on CMMs.

4.2 Tolerancing Methods and Standards
Tolerances, as defined by the British Standards Institution, are (BS 308 Pt2 1985):

"The total amount of variation permitted for the size of a dimension, a positional relationship or the form of a profile or other design requirement".

Dimensional and geometric tolerances provide the designer with a specification in which he or she can control the functionality, interchangeability and manufacturability of a product. As the assignment of tolerances can have a
considerable influence on the cost and quality of product manufacture, the correct selection of the appropriate tolerances is an essential activity in the design process. Therefore, geometric and dimensional tolerances are of fundamental importance in the construction of any product model definition. Although conventional linear and angular dimensions and tolerances have been employed since the beginning of the century, it was only after the advent of the Second World War that efforts were made to standardise tolerancing practices (Hill et al 1976, Corrigall 1990, Voelcker 1997). These efforts have resulted in the creation of a number of standardisation documents for controlling the representation of dimensional and geometric tolerances on mechanical engineering drawings. The two principle standards most commonly used are:

i) British Standards Institution - BS308;
ii) American National Standards Institute - ANSI Y14.5M.

Voelcker (1997) provides a summary of the evolution of mechanical tolerances from a historical perspective, the current state of tolerancing technologies and the recent surge of research directed at rationalising and mathematising form tolerancing.

### 4.2.1 British Standard Institution's BS308: Engineering Drawing Practice

This standard was prepared under the direction of the Mechanical Engineering Standards Committee to provide recommendations for the general principles of presentation and practice to be applied to all engineering drawings. BS308 was first introduced in 1927 and is currently at its fifth revision. The standard is comprised of three major parts (BS 308 Pt.1 1984, BS 308 Pt.2 1985, BS 308 Pt.3 1972):

i) **BS308 part 1**: Recommendations for general practice (corresponds to the International Standards Organisation (ISO) standard ISO128);

ii) **BS308 part 2**: Recommendations for dimensioning and tolerancing of size (corresponding to ISO standards No. 129, 406 and 1302);

iii) **BS308 part 3**: Recommendations for geometrical tolerancing (corresponding to ISO standards No. 7083).
4.2.2 American National Standards Institute: Dimensioning and Tolerancing

ANSI Y14.5M

This standard was firstly introduced by the American National Standards Institute in the December of 1982 and was revised by the American Society of Mechanical Engineers (ASME) in 1994 (ASME Y14.5M 1994). Its main objective was to co-ordinate and integrate dimensioning and tolerancing techniques into and via the computer and any other electronic data systems for design, manufacture, verification and similar processes. This standard provides a comprehensive document defining engineering drawings and related documentation practices and covers the following subject areas:

i) Definitions and General Dimensioning;
ii) General Tolerancing and Related Principles;
iii) Symbology;
iv) Datum Referencing;
v) Tolerances of Location; and
vi) Tolerancing of Form, Profile, Orientation and Runout.

The standard identifies that the size and shape of a manufactured component can vary from the perfect or nominal form in a variety of ways and categorises these variations into six standard tolerance classes (ASME Y14.5M 1994):

i) Size; iv) Position;
ii) Form; v) Profile; and
iii) Orientation; vi) Runout.

Despite the combined efforts of the standard committees, the representations specified in these standards are complex and suffer from ambiguities, inconsistencies, and are open to misinterpretation (Corrigall 1990, Feng and Yang 1995).

4.2.3 The Importance of Following Geometric, Dimensioning and Tolerancing Specifications

As mentioned previously, the use of geometric, dimensioning and tolerancing (GD&T) was primarily employed as the tool in which the designer’s intent could be expressed and conveyed. From the basis of the specified dimension and tolerance defined by the
designer, process engineers could select the appropriate processes and plan the manufacturing operations. Quality engineers examine the product's dimensional conformity by comparing inspection data with tolerance specifications (Ge et al 1992). There are however, additional benefits that can be realised through the implementation of a combination of GD&T symbols, collectively known as a GD&T specification, on engineering drawings (Wearring and Karl 1995):

i) GD&T specifications can provide the documentation base for the design of both quality and production systems; and

ii) GD&T specifications promote a uniform interpretation of a component's production requirements.

Wearring and Karl demonstrated, with a computer simulation of the fabrication of five thousand simple workpieces, the negative effects that could be encountered if GD&T specifications were ignored. During the simulation each component was manufactured, inspected and assembled using a specialised inspection gauge. The results of the simulation indicated that the percentage of components passing the inspection was approximately 99% when the GD&T specification was followed. However, acceptability dropped to approximately 20% when both improper component positioning and feature inspection was employed. As all the results were obtained from the same batch of five thousand components, i.e. the 79% of acceptable parts were rejected in the latter simulation, the increase in rejection rate was attributable to the positioning procedure of the component and the sequence used during inspection.

4.2.4 The Addition of Tolerance Information to Solid Models

Although there have been several attempts to standardise the dimensioning and tolerancing representations, these schemes have focused on a combination of both textual syntax and symbolic semantics on engineering drawings. The main drawback in specifying symbols on drawings is that computer-aided systems, that engage in GD&T analysis, cannot directly interpret or manipulate this type of representation (Feng and Yang 1995). Although the DMIS specification (Audin 1987, ANSI/CAM-I 1990) specifies geometric tolerances in accordance to ANSI Y14.5M, these representations are purely limited in the scope of inspection and do not support the GD&T relationships for other product life cycle applications.
Current solid modellers, although they contain complete and unambiguous representations of nominal or perfect component geometry, suffer from severe deficiencies in incorporating tolerance information (Requicha 1983, Ge et al 1992, Menq et al 1992, Feng and Yang 1995). This deficiency forms one of the major integration obstacles of implementing a computer integrated manufacturing system.

An early and comprehensive study into the representation of tolerances within solid models was reported by Requicha (Requicha 1983, Requicha and Chan 1986). Requicha identified that the incorporation of tolerances into solid models raised a number of key issues that needed to be addressed. These issues could broadly be categorised into three areas:

i) representation of tolerances;
ii) analysis and synthesis of tolerance specifications; and
iii) application of tolerance information.

Requicha’s solution involved the use of a variational geometry concept. This concept consisted of envelopes of geometry of similar configuration to the feature it represents. Tolerance zones were created by pairs of offset envelopes that enclose the original feature shape. The inner and outer boundary of the tolerance zone respectively represent the Least Material Condition (LMC) and Maximum Material Condition (MMC) of the feature (ASME Y14.5M 1994). This tolerance zone also corresponds to the maximum variation from nominal form allowable during manufacture. These tolerance zones were stored as variational geometry in the form of a Vgraph representation along with the solid model representation. Feng and Yang identified that this methodology was best suited for tolerance zones that have fixed positions and not where tolerance zones have floating positions, such as form and orientation tolerances.

Another method of representation of dimensional and geometrical tolerances adopted by many researchers is the use of the attribute facility that is available with most solid modellers (Menq et al 1992, Walker and Wallis 1992). The inspection specification module proposed by Menq et al employed the attribute function of the CATIA geometric modeller to assign non-geometric information to the part model features. Walker and Wallis at Bath University developed a solid model inspection system in which the tolerance attributes were assigned to a particular feature by the use of three input language functions:
i) \textit{datum()} - this function was used to define a surface as a datum;  
ii) \textit{tol\_set()} - this function was used to define a tolerance; and  
iii) \textit{attribute()} - this function was used to define inspection auxiliary information.

Although this method provides a solution for the representation of tolerance information in solid models, it does not address the initial problem of the serious dimensional and geometric tolerancing deficiency of current solid modellers.

4.2.5 Tolerance Modelling

Roy et al (1989a, 1989b) developed a tolerance representation scheme for application within a solid model environment. This approach was based on the Quick Turnaround Cell Design System (Chang et al 1988) and the TWIN solid modelling package developed at the Purdue Engineering Research Centre (ERC) at Purdue University. This system possesses the capability of the augmentation of the tolerance representation within either the unevaluated CSG or the evaluated B-Rep solid models. The effective linking of both CSG and B-Rep data models was achieved through the development of a reference face list. Tolerance information was attached to the reference face list as constraint nodes (Roy et al 1989b). This contribution was achieved through the development of three elements (Roy et al 1989a): an efficient user interface system for the interactive input of tolerance information, a suitable tolerance data model interfaced to the solid modeller; and an information retrieval system for the stored tolerance information for subsequent analysis.

The Expert Programming System – One (EPS-1) operational environment developed at the University of Texas (Reimann and Sarkis 1993) employed a specially developed dimension and tolerance modeller that defines tolerance nodes and assigns the nodes to one or more geometric features. After the addition of geometric features to a part model the D&T modeller augments the geometrical entities with the specific tolerancing information. Another concept for the incorporation of geometrical tolerances involved the use of explicit data structures, known as objects, that capture the GD&T information in the form of a tolerance model.

One such system reported by Feng and Yang (1995) was the development of a representation scheme for a dimension and tolerance data model. The model was conceived to enable tolerance specifications to be associated with solid models and to facilitate exchange of D&T information to other applications within a heterogeneous
computing environment. Formulated in the EXPRESS product data modelling language (Owen, 1993) the main objective of the model was to:

i) convey the functional requirements from design to manufacturing in a computer compatible format;

ii) define a common semantic format for representation and exchange of dimensioning and tolerancing information;

iii) be an integral part of the standard product data model;

iv) associate D&T data with CAD to replace traditional paper drawings;

v) provide a computer interoperable data format for using D&T data and communicating it between CAD, manufacturing, tolerance analysis, tolerance synthesis, assembly analysis, process planning and inspection planning systems.

The resultant model consisted of three groups of EXPRESS schema: a dimension schema; a tolerance schema; and a datum and shape aspect schema.

A Tolerance Modelling System (TMS) was proposed by Kulkarni and Pande for the representation and analysis of manufacturing tolerances using solid models (Kulkarni and Pande 1992). The TMS was designed with five governing characteristics in mind:

i) **completeness** - capable of handling all types of features;

ii) **validity** - assuring the validity of tolerances, removal of redundant tolerances;

iii) **adherence to standards** - the reasoning that drives the systems must be derived from international tolerancing standards;

iv) **feature handling capability** - support for both feature-based and low level abstraction in the specification of tolerances; and

v) **open structure** - must provide open access to its data structures that provide both feature as well as tolerance information and their interpretation.

The TMS was based on object oriented principles and focused on issues of automatic feature recognition, feature data organisation for tolerancing, tolerance information structure to represent datums, and tolerance types. The modelling system provided the facility for tolerance data storage, retrieval and analysis. The TMS comprised of two sub-systems: a feature subsystem; and a tolerance sub-system and was
built around a constructive solid geometry (CSG) and a boundary representation (B-Rep) hybrid solid modeller. The structural layout of the tolerance modelling system consists of the following:

i) **Feature subsystem** comprised of a set of programs that perform specific tasks of automatic manufacturing feature extraction, recognition, classification and organisation of functionally relevant features from the solid model.

ii) **Tolerance subsystem** which interacts with both the feature subsystem and the user. Its purpose was to obtain, validate, store and interpret tolerance related data. Data was organised as a collection of classes whose objects interact with each other to represent both feature and tolerance information. Communication was achieved through the issuing of messages. Four object classes were identified: feature class, abstract entity class, tolerance class and datum class. The tolerance interpretation module ensures the validity of the proposed tolerance scheme and provides meaning to the information stored in the data structure of the TMS.

Shah et al (1998) conducted a study into the application of a GD&T model for use with both design and process planning phases of the manufacturing cycle. The GD&T model was employed initially to capture the design intent of the designer, whilst complying with the ANSI Y14.5M standard, within a functional feature design model. Manufacturing or machining features are automatically extracted from the function feature design model through feature recognition. These manufacturing features are appended with the designer's GD&T scheme to provide an appropriate format for the CAPP activity. The GD&T scheme is required by the CAPP systems to determine from the manufacturing features, the type of operations to use and the sequence of execution of the operations. The GD&T semantics in the functional design features differ from the meaning of the GD&T representation associated with the manufacturing features. The design feature GD&T specification represent the sizes and shape of the features the constitutes the part, whilst GD&T associated with manufacturing features represent the sizes and the maximum size and shape variations of the volumes to be removed from the initial stock material.
4.2.6 Critique – Tolerancing Methods and Standards

Tolerancing standards have provided an invaluable vehicle to convey information from the designer to the planner and operator on how to control the functionality, interchangeability and manufacturability of a product. However, there are many possible tolerance schemes in which a designer can express the functionality of a product. This ambiguity, inconsistency and complexity of possible tolerancing schemes coupled with the fact that they were primarily developed for use with manual engineering drawings, makes it difficult or impossible for a computer systems to interpret all possible connotations.

GD&T symbolic nature provides a mechanism which attempts to simplify this problem as it allows tolerance verification checking to take place thus preventing the specification of tolerances that rely on non-existent datums. This representation possesses another problem as it requires additional interpretation from the computer in order to convert the symbolic representation into a computer interpretable format to facilitate subsequent analysis.

It is because of this complexity the study of tolerance modelling is wholly practised in the academic arena whilst in the industrial software vendor community the application of tolerance modelling proves to be an impenetrable barrier against CAM and CAI integration.

4.3 Inspection Planning for CMMs

This section investigates the contemporary research into the automated inspection of manufactured components using CMMs. The research area reported is mainly concerned with exploiting the potential offered by the computer controlled CMM as an effective process feedback and process control device. The areas investigated include:

i) Current CMM Operating Practices;
ii) Feature Recognition and Inspection Requirements Specification;
iii) Measurement Planning and Sequencing;
iv) Inspection Plan Simulation and Verification;
v) Inspection Plan Execution.
4.3.1 Current CMM Operating Practice

About 20 years ago the introduction of manual CMMs brought about drastic reductions in inspection times. Typically a 90% saving was observed over the traditional methods involving height gauges, dial indicators and surface plates (Bosch 1991). With the advent of Direct Computer Control (DCC) CMMs, to inspect manufactured components comprising of complex configurations, these astounding reductions have been enhanced dramatically (Harris et al 1994). Harris et al has quoted a 90% time saving can be observed with the adoption of computer directed inspection as opposed to a manual joystick driven CMM. The potential benefits to be exploited through the adoption of a DCC CMM can be characterised into six main categories (Shaffer 1982, Gu 1994, Elmaraghy and Elmaraghy 1994):

i) **Flexibility** - CMMs provide the capability to measure any dimensional characteristics at any configuration;

ii) **Reduction of set-up time** - The operator only requires to position the component in a convenient orientation onto the CMM table. All the co-ordinate data is compensated automatically according to the calibration of the component and the probe;

iii) **Improved accuracy** - As all the measurements are conducted from a common measuring system the accumulation of errors resulting from traditional inspection methods is eliminated;

iv) **Speed of execution** - Allowing more frequent sampling of points possible;

v) **Statistical process control (SPC)** - can utilise the results data to ascertain process capability and determine the onset of unacceptable process variability at an early stage. This would facilitate the application of timely correction measures to ensure process control reducing costly scrap and rework;

vi) **Automation** - As CMMs possess computer controllers that utilise high level programming languages on the one hand, and coupled with the aforementioned benefits on the other, they are considered as the perfect automated inspection tool.

However, in almost all cases, these expensive and advanced measuring machines are operated manually by operators based on their experience and interpretation of the component inspection criteria (Gu 1994). This has had the effect of relegating these
sophisticated and flexible machines into a low level automation role. To achieve the goal of automated inspection within a CIM environment the activity of CAD-based inspection planning forms the foundations of any efficient automated inspection system. Such a system would incorporate: feature recognition; inspection requirements specification; measurement planning and sequencing; tolerance planning; probe selection and orientation; off-line programming; and simulation.

4.3.2 Feature Recognition and Inspection Requirements Specification

A pioneering attempt to reap the potential benefits offered by CMM inspection was the generative PROLOG based expert inspection task planning system developed by Elmaraghy and Gu (1987) at the McMaster University of Canada. This system was created to automatically generate inspection plans for rotational components. The issues addressed by this approach included CMM characteristics, inspected part function and geometric properties, geometric tolerancing theory and feature extraction. A technique of syntactic pattern recognition was employed to identify the part directly from the feature-based CAD database. A feature-oriented modelling system was developed comprising of three main constituents: a feature base; an interactive dimensioning and tolerancing module; and an expert tolerancing consultant. The feature base was created through the examination and decomposition of a representative sample of parts to establish feature relationships and their associated manufacturing operations required to produce them. Both dimensional and geometric tolerances are interactively allocated to the individual features of the component model with the aid of the expert tolerancing consultant that can assist in the identification of the optimal geometric tolerances. All the data pertaining to the machining operations and geometric tolerances are captured in a part data file created by the feature based modelling system, which forms the basis of the inspection planning system.

Research undertaken at UMIST resulted in a knowledge-based process planning system for rotational parts, known as Expert Computer Aided Planning System (EXCAP) (Tang and Davies 1990,1995). The system was capable of automatically generating process plans from its own knowledge base and the geometry obtained from an IGES file representation generated from the CAD system. INSPEX is a knowledge-based inspection planner that augments the EXCAP knowledge-base with inspection rules for the generation of plans for both in-process and post-process inspection of turned components (Kalta et al 1992, Tang and Davies 1995). The initial data
requirement for the INSPEX planner could be in the form of an IGES file representation of a toleranced product model from the CAD system or the partially processed product model produced during planning execution of EXCAP.

The aforementioned approaches have concerned themselves with the feature extraction from two-dimensional CAD models and are limited to rotational type components. However, the generative Inspection Process Planning Expert (IPPEX) created by Brown (1991) and the EPS-1 architecture developed at the University of Texas (Reimann and Sarkis 1991, 1994) adopted a twin model method comprising of a geometric modeller and a dimensional and tolerancing modeller. The combination of both the geometric and tolerance models constructed by the modellers provided a complete product model definition of a three-dimensional component. The retrieval of the geometrical and tolerance information was achieved by interrogation with the models through the IGES like applications interface and the dimensioning and tolerancing applications interface specifications respectively.

An intelligent dimensional inspection environment conceived at the Ohio State University consists of five functional modules (Menq et al 1989, Yau and Menq 1992, Ge et al 1992):

i) *inspection specification module* consisting of dimensional and tolerance (D&T) procedures that allow the user to assign tolerances to the component’s geometrical model, specify machine constraints, and special customer requirements. This non-geometric information is attached to the geometric model as attributes and forms the basis of an inspection specification;

ii) *automatic inspection planning module* which processes the inspection specification and determines the number of probing points, probing vectors and automatically produces an inspection plan;

iii) *CMM verification module* to ascertain the acceptability of the initial inspection plan by detecting any probe and component collisions through simulation. Any path planning problems encountered are rectified at this stage;

iv) *CMM execution module* consisting of two parts, part alignment and component inspection;

v) *comparative analysis module* to analyse the inspection results and decide on the acceptance or rejection decisions based on the principle of hypothesis testing.
The inspection specification module follows a similar methodology as the expert inspection planning system proposed by Elmaraghy and Gu (1987). Geometric Dimensioning and Tolerancing (GD&T) procedures have been implemented within the CAD system. These procedures provided the designer with the ability to assign tolerances and other non-geometrical information, such as special customer requirements, machine capabilities, CMM constraints, to the three-dimensional CAD model. The non-geometric data elements were stored as attributes of the feature in the part model through designer interaction via the CAD system's graphic interaction interface. These attributes were processed by the specification module to produce the inspection specification which was then used to drive the automatic inspection planning module. One drawback of this approach was that the CAD system employed was not a feature-based system and as a consequence of this the feature type had to be assigned to each feature at the same time as the non-geometric information.

The intelligent interface link between a CAD system and a CMM contrived at Brunel University (Mullineux 1988, Singh et al 1990, Hassan et al 1992, Medland and Mullineux 1993a) comprises of a distributed system in which the inspection task was controlled by a combination of dedicated programs and the constraint modelling system RASOR (Rules And Systems Of Rules). An extended feature file was constructed from the interrogation of the CAD database and a number of other sources. The data contained in the feature file provides all the necessary information required for measurement execution on a CMM and is categorised into five basic types namely:

i) geometric feature description;  
ii) additional probing geometry;  
iii) approach geometry;  
iv) tolerance conditions; and  
v) control parameters.

4.3.3 Measurement Planning and Sequencing

Early analysis of traditional inspection processes, undertaken by Elmaraghy and Gu (1987) identified three generic stages essential in order to produce an inspection plan. These stages consisted of:

i) the correct interpretation of the design criteria as specified in the drawings or the CAD model;
ii) the formulation of strategies regarding the inspection procedure to be followed in the light of the available inspection facilities; and

iii) the CMM instructions required for inspection plan execution.

Over the past decade, most automatic planning research into the dimensional and geometrical inspection of manufactured components have concerned themselves with either (Requicha 1990, ElMaraghy and ElMaraghy 1994):

i) high level planning tasks such as set-up determination with relation to accessibility of inspection features, optimisation of the inspection procedure in line with some predetermined criteria; or

ii) lower level planning activities which address issues of point selection, path generation / simulation and machine code generation.

One of the earliest problems encountered with research into the creation of an expert inspection planner for multi-component inspection addressed at both the McMaster University and The University of Calgary, Canada (Elmaraghy and Gu 1987, Elmaraghy and Elmaraghy 1994, Gu 1994, Chan and Gu 1993) was the identification of those geometric elements that were difficult, or impossible to be inspected on a CMM. For example, components such as gear tooth geometry and screw thread profiles are best suited to inspection on special purpose optical projectors. This approach frees the CMM from inefficient and time consuming inspection practices.

4.3.4 Probe Configuration and Orientation Selection

To inspect a component on a CMM equipped with a tactile touch probe, the accessibility of a component’s surface features is vitally important for the determination of probe orientations and trajectories that are required for the creation of a collision free inspection path. Requicha and his colleagues at the University of Southern California, whilst working on an NSF funded project to extend the computational capabilities of solid modellers, identified that the planning for component inspection on a CMM included the following tasks (Requicha 1990, Spyridi and Requicha 1990):

i) selection of workpiece orientations;

ii) selection and placement of fixtures and clamping devices;
iii) machine selection;
iv) probe selection;
v) sample point generation;
vi) probe trajectory generation;
vii) generation of the servo commands for the CMM controller.

The architecture of their prototype inspection planner is comprised of the aforementioned high and low level planners coupled with a servo command generator. The proposed high level planner consisted of a number of specialists that produce a partially-ordered collection of set-ups, and a partially ordered set of operations at each set-up. They proposed an accessibility specialist to provide a solution to feature accessibility problems, which are crucial in the selection of orientations for both the workpiece and the probe. A probe abstraction consisting of a half line with an end point, was employed to determine the orientation of the probe that can inspect a feature without interfering with the workpiece which is known as the accessible direction for the feature (Spyridi and Requicha 1990). Each of the directions corresponds to a probe orientation and, in the case of straight probes, also determines the orientation of the workpiece relative to the CMM. A set of accessible directions possessing a common end point was referred to as an accessibility cone. Their proposed methodology consisted of two stages, accessibility analysis and clustering. Accessibility analysis involved the computation of all the directions for each feature of interest. Clustering was the process of ascertaining a minimal set of directions to enable the inspection of all the features. Spyridi and Requicha distinguished between both the local and global accessibility of a measurement point:

i) *local accessibility* was concerned with obstacles within the immediate locality of the point being measured. 

ii) *global accessibility* was concerned with the entire workpiece.

Local accessibility cones (LACs) on their own did not guarantee that a surface feature is accessible by the probe and therefore, the computation of the global accessibility cones (GACs) was required. As GACs are a sub-set of LACs and consist of all the probe directions that can access the feature without interference. This approach was only capable of producing a sub-optimal solution and as a consequence of
a simplified probe representation employed in the LAC and GAC computations it could only be used to determine what surface features were inaccessible by the real probe. Zeimian and Medeiros (1998) incorporated this accessibility analysis technique in their probe selection and part set-up planner for the inspection of prismatic components on CMMs.

The research work undertaken by Spyridi and Requicha was restricted to quadratic and planer surfaces and only described the calculation of the LAC of the entire feature surface. Menq et al at The Ohio State University extended this work to include the accessibility analysis of sculptured surfaces through the computation of the GAC using an enhanced probe abstraction. This system was successfully implemented with the verification module which formed part of the their intelligent inspection planning environment (Yau and Menq 1991, 1995, Menq et al 1992, Lim and Menq 1994).

An alternative probe and component set-up planner which forms a part of an inspection plan and code generator was researched by Corrigall (Corrigall 1990, Corrigall and Bell 1991). This project was nationally funded by the Engineering and Physical Science Research Council (EPSRC) on the Information Support Systems (ISS) for design and manufacture research programme, which was undertaken as a joint venture between the University of Leeds and Loughborough University. The probe and component set-up planner constituted one of three planners employed by the inspection plan and code generator facility:

i) the operation type planner;
ii) the probe and component set-up planner; and
iii) the operation data planner.

The probe and component set-up planner involved the creation of probe approach directions (PADs) from the constructive solid geometry (CSG) representation of the component using a method based on the spatial decomposition technique. The component’s bounding area was decomposed into small cubic cells which were approximately the size of the probe’s tip. The cells were designated as one of three types:
i) **full cell** - a cell that is totally enclosed within the object;

ii) **empty cell** - a cell totally outside the object; and

iii) **boundary cell** - a cell on the objects boundary.

The boundary cells were used to create a surface cell list to analyse the PADs. The probe movement envelope (PME) was modelled as being the swept volume occupied by the moving probe during point measurement, much in the same way as a cutting tool motion is modelled for machining processes. The amalgamation of a number of PMEs to cover the entire surface of interest is termed the total probe movement envelope (TPME). If the TPME interferes with any part of the surface then that area of the surface is deemed unapproachable by the probe and the coverage of the offending PAD is reduced accordingly. The set-ups were determined by generating a list of all possible principle component orientations and ascertaining the minimum combination of these that will allow all of the tolerance assignments to be inspected. The component orientations with the largest number of probable features was selected first and the process was repeated until all the tolerance assignments have been allocated to a particular component orientation. Each component set-up specification contained a list of probe configurations, orientations and operations that were associated with each component set-up. The methodology conceived by Corrigall was based upon three underlying assumptions namely:

i) any surface can be probed in at least one of six approach directions based on the component’s axis system;

ii) simple fixturing methods are used whenever possible as the incorporation of a fixture design facility is unavailable; and

iii) a single probe configuration using a sphere type stylus is used throughout the inspection cycle.

**4.3.5 Probing Point and Path Planning**

Much research has concentrated on determination of the probing points and their sequence used to measure particular features. Atkins and Derby (1989) whilst conducting research into product data definition developed an inspection program generation facility for Pratt and Whitney. Their inspection point and path planning was achieved by a combination of two methods. Simple feature inspection points and path
creation were generated through interactive parameterisation of a set of macros. For more complex features the points and path were achieved manually. One drawback of this approach was that the sequence planning was left to the discretion of the operator.

A similar approach has emerged from efforts to develop a Rapid Design System (RDS) to reduce the time from design to manufacture and inspection (Merat et al 1991, Merat and Radak 1992). In this work a high level representation of a measuring method for a given GD&T feature, known as inspection plan fragments (IPFs), were created. Inspection code fragments (ICFs) were generated based on the IPF for each inspection feature and consisted of DMIS codes segments, probe start and retract positions. These fragments were combined to produce DMIS inspection code for CMM execution. The plan sequencing was based on the minimum number of component set ups as is the case with the inspection plan and code generator proposed by Corrigall.

Chan and Gu decomposed the inspection planning task into a multi-module knowledge based inspection planner that was based on object oriented principles (Chan and Gu 1993, Gu 1994, Gu and Chan 1995, 1996). The resultant object oriented knowledge based inspection planner (OOIP) utilised the concept of passing messages between objects defined within the system as the data transfer mechanism from module to module. Their procedure consisted of five modules: manual and machine module; accessibility determination module; datum searching and sequencing module; probe selection module; and the assignment of the number of measuring points module. Within the accessibility module, features with the same accessibility are grouped together and assigned to a probe configuration. The datum searching and sequencing module attempts to minimise the number of probe changes. Probe sequence selection was achieved on a datum priority basis. For example, a feature list of a particular probe configuration contains a feature item that requires a reference datum. However, if the measurement of that reference datum is contained in the feature list of another probe configuration then the latter possesses a higher priority than the former and is subsequently placed within the inspection plan first. The number and position of inspection points for each feature was dependent on the data fitting algorithms employed for the inspection analysis. Medland et al (Hassan et al 1992, Medland and Mulli Xuex 1993a) at Brunel University also adopt the feature priority approach which allows the inspection to be aborted upon error detection and parts to be rejected without wasting time on the measurement of less critical features.
All the research contributions mentioned above have neglected to address the complex issue of inspection point distribution. Corrigall recognised that in an ideal world the probing points should be distributed evenly so that the distance between neighbouring points should be equidistant in order to generate the most accurate representation of the surface. Medland and his team at Brunel University identified the need to recast tolerance conditions for use with CMMs (Medland et al 1994) which employed a sample-grid procedure to create a grid of points across the modelling features during their construction within the CAD system. Menq and his colleagues have conducted an extensive study into the statistical evaluation of form tolerances for free-form surfaces. Their proposed scheme determines a suitable probing point sampling plan that is capable of representing the entire population of possible probing points to a sufficient confidence level (Menq et al 1990, 1992b). An optimal match procedure establishes the best fitted features using a non-linear least squares method. Using the sample size and the optimal match fitting approach the calculated deviations are used to ascertain acceptability status of a manufactured feature via hypothesis testing. Pahk's work on the automation of the inspection of dies and moulds identified the need for a number of probe point sampling strategies and proposed three types of sampling strategies (Pahk et al 1993, 1995):

i) **uniform distribution** - a grid is formed by the measurement points have a nearly square distribution which gives a uniform coverage over the sculptured surface. BS 7172 (1989) provides a guide to CMM manufacturers and software writers in the assessment of position, size and departure from normal form of geometric features and stipulates the minimum number of uniformly distributed probing points for primitive type features.

ii) **curvature dependent distribution** - in this method of distribution the number of sampling points increases as the degree of curvature increases; and

iii) **hybrid distribution** - the distribution of measuring points was determined as a combination of both the uniform and the curvature distributions and was employed to avoid too many measurement points being concentrated in regions of high curvature.
Fan and Leu (1998) formulated a CAD-directed inspection planning system for
CMMs that employs an algorithmic approach for analysing and specifying uniformly
distributed probing points for planes, cylinders and cone type features.

Once the sequence of inspection features and their associated probing point
locations have been determined the final process involves the joining of these features
together via a collision-free path to formulate the initial inspection plan.

4.3.6 Inspection Plan Simulation and Verification
Just as NC verification is vitally important for any machining operation, inspection plan
verification is of equal importance for automatic inspection, especially if the CMM is to
be driven under direct computer control. The traditional method of programming a
CMM was achieved by the on-line teach and playback approach that required the
CMM, component and a computer terminal to produce an inspection program. With the
advent of the integration of CAD systems and CMMs the full potential of off-line
programming could be exploited inasmuch as the whole programming process could be
executed on a CAD/CAI system, through planning to simulation and plan verification,
thus freeing the CMM to perform more productive tasks. Early research undertaken by
Cowling and Mullineux (1989) into the integration of CAD and CMM equipment
employed a simulation of the CMM to verify the feasibility of their intelligent interface.

The Inspection Process Planning Expert (IPPEX) (Brown 1991) and the Expert
Planning System - One (EPS-1) (Reimann and Sarkis 1991, 1994) systems both utilised
simulation of the inspection probe path through a simplistic representation of the
probe's tip and axis in the form of two parametrically synchronised curves. The probe
path is animated graphically on a graphics user interface for the planner to review. One
major disadvantage with this method is that collision detection is the responsibility of
the planner.

A more comprehensive approach of simulation and verification was employed in
the CMM verification module (CVM) of Menq et al's automated dimensional
inspection environment (Menq et al 1992, Yau and Menq 1992, 1995). The initial
inspection path created within the automatic planning module did not constitute a
perfect solution and consisted of defects such as interferences or collisions between the
part and the probe. In this work, a hierarchical planning system using heuristic
modification was established to overcome this planning problem and consisted of four
main planning features namely:
i) a 3D planning system;

ii) collision detection and modification of the trajectories of the probe's tip, stylus and column;

iii) construction of a CMM model with selectable parameters for geometric simulation;

iv) dynamic simulation to reveal the real CMM operation and generate the servo command for its control.

Although the proposed automatic dimensional inspection environment conceived by Menq et al is one of the most extensive research contributions into inspection planning systems, it does not address the issue of optimal inspection sequencing. Moreover, the scope of application is restricted to components comprising of complex surfaces.

Fan and Leu's (1998) approach to inspection probe path planning and collision detection utilised a swept volume methodology, similar to the total probe movement envelope (TPME) adopted by Corrigall and Bell (1991), to detect probe and workpiece collision points. Potential collisions are avoided through either a modification to the probe's attack angle or by modifying the probe trajectory by the inclusion of an additional movement point.

4.3.7 Inspection Plan Execution

Prior to generating the CNC commands for a 3-axis CMM, a component alignment computation must be performed (Pahk et al 1993, 1995, Yau and Menq 1992). Part localisation for prismatic components could be achieved by the traditional 3-2-1 method that involved taking three measurements on a primary plane, two measurements on a secondary plane, and one on a tertiary plane. However, for objects possessing complex sculptured surfaces the traditional alignment methods were slow and inconsistent as they required expensive and specialised fixtures to be made. Menq and his associates proposed a CAD-model-based localisation algorithm that is capable of overcoming the difficulty of mathematically locating the component comprising of sculptured surfaces (Menq et al 1992, Yau and Menq 1992). With the first of these two methods, a transformation matrix was computed and applied to compensate during the inspection process. The completed inspection plan can be either be post-processed into CMM dependent machine code, such as CMES (Harris et al 1995) and HTBasic (Ferranti
Metrology Systems 1986) or pre-processed into the neutral data transfer format DMIS (Tang and Davies 1995, Brown 1991, Kalta et al 1992, Merat and Radack 1992, Klages and Wilson 1994) (see chapter 3.5.3.1). These generated inspection programs are then downloaded onto the CMM for automatic execution under computer control. An alternative to the production of an inspection program is to directly control the servo motions of the CMM in accordance with the inspection plan for a computer via a machine tool interface. This type of control is adopted by the commercially available Valisys™ inspection module (Tecnomatix Technologies Ltd 1995). The measurement data obtained from the execution of the inspection task is subsequently stored awaiting examination by the comparative analysis module.

4.3.8 Critique – Inspection Planning for CMMs

Although the activities involved in automated inspection planning can be considered closely related to the operation planning on the process planning activity, they are treated as mutually exclusive and therefore researched and developed in isolation by both academic institutions and industrial software vendors alike. This isolation has led, in the same way as the interest in operation planning, to a plethora of fragmented solutions that focus on very specific issues within a limited application domain. These embryonic solutions, whilst being significant contributions, are very complex and specific in nature. This complexity of inspection planning coupled with the other complex research interests, i.e. product modelling, manufacturing resource modelling, machine and operation planning etc., not to mention standardisation issues, are providing a major obstacle to the integration of the design to manufacture phases of the product life-cycle.

4.4 Inspection Results Evaluation

This section reviews the current research and issues being addressed in the pursuit of an automated, integrated and flexible computer aided inspection (CAI) system. The areas investigated include:

i) Requirements for Research in CAD Based Automated Inspection;
ii) CAD/CMM Integration;
iii) Integration through Data Exchange Standards;
iv) Inspection Integration into Automated Manufacturing;
v) Automated Inspection Environment;
vi) Inspection based on CAD Models; and
vii) Emerging Research Issues in the Context of CAD-directed Inspection.

4.4.1 Requirements for Research in CAD Based Automated Inspection

Although the automation of many manufacturing activities has been well researched and understood, the philosophy of CIM attempts to link these 'islands of automation' allowing interactions between activities through the exchange of information. One such link that is thoroughly established through the implementation of feature-based design technologies is the integration of CAD/CAM systems (Walker and Wallis 1990, Shah et al 1991, Shah 1991, Salomons et al 1993, Case and Gao 1993). Until recently the link between CAD and inspection systems has received little attention from research establishments. However, as a result of the pressures exerted by the customer for better, faster and cheaper products, research issues regarding the interfacing of CAD and quality systems need to be addressed if increased production efficiency and product quality is to be achieved.

4.4.2 CAD/CMM Interfacing

Example implementations of automated inspection systems within manufacturing have been around for a number of years (Treywin and Edwards 1987, Cardew 1987). However, these systems have evolved in the aerospace and automotive industries and comprise of sophisticated, specialised and dedicated arrangements. Such systems lack the flexibility to cope with product variety and the real-time requirements experienced within a contemporary small batch manufacturing environment. With the introduction of CMMs into manufacturing, both flexibility and consistency can be greatly enhanced by interfacing such machines with CAD systems. It is clearly advantageous to employ the component's CAD model as the standard to which the produced component can be compared and assessed (Cowling and Mullinium 1989).

Early research into CAD-directed inspection of manufactured components on a multi-axis CMM was undertaken at the former National Bureau of Standards (Hopp 1984). The main objective of this work was the identification of the enhancements required by existing CAD databases to support quality assurance related information, from which inspection procedures could be automatically generated. Hopp proposed a control system hierarchical architecture that employed task decomposition techniques to
partition the global goal of complete component inspection into a set of individual sub-tasks which could be executed sequentially on a CMM. Hopp recognised seven hierarchical levels of control that can be applied to CMM component inspection:

i) **tolerance** verification selection;
ii) **feature** selection to verify a particular tolerance;
iii) **surface** selection to measure the feature;
iv) **probing points** required to measure individual surfaces;
v) **probing paths** to measure each point;
vi) **machine motions** required to move through the probing path; and
vii) **servo commands** needed to produce the machine motions.

An early attempt into the linking of the design and inspection activities was reported by Cowling and Mullinuex (1989) at Brunel University which resulted in an two-way intelligent CAD-CMM interface for use in the inspection of 2.5D components. The main purpose of which was to alleviate the CAD system and user from manipulating and interpreting large amounts of measured data. The interface provided the capability of: dealing with part geometry at the entity and feature level as opposed to the data transfer of measured raw point data; interrogating the CAD model database; determining whether or not manufacturing errors exist; and altering the overall inspection strategy based on the findings.

The above interface was improved with the introduction of the artificial intelligence concept of the Blackboard. This concept was utilised to provide an intelligent bi-directional communication interface between a CAD system and a CMM (Singh et al 1990). The blackboard is analogous to a formal committee meeting attended by several experts and governed by a manager or chairman. The experts, the CAD system and the CMM, indulge in a two-way communication monitored and controlled by the manager/chairman represented here by the blackboard. The system was based on an existing constraint modelling system called RASOR (Mullinuex 1988). The information within the blackboard was held in a hierarchical arrangement and is so structured to allow graphical representations from both the CAD system and CMM.

The blackboard system was limited to the bi-directional communication between two personal computers (PCs) via a serial connection and only allowed one activity to be performed at any given time. A more flexible approach evolved from the blackboard
concept which consisted of a suite of programs that facilitate the communication within a multi-tasking environment, i.e. the CIMITER program to extract CAD model data from the part file, and an EXCHANGE program to allow the transfer of information (Hassan et al 1992). The control of the process throughout the whole inspection activity was achieved by the RASOR constraint modeller.

4.4.3 Integration through Data Exchange Standards

All of the systems mentioned up until now have been tailored to specific CAD systems and CMMs. For ultimate flexibility many systems have been adopted to use a data exchange approach for the interfacing of design and inspection computerised equipment. Research has been conducted at the University of South Wales, Australia (Farmer and Smith 1991) that consists of a PC-based inspection system that allows product specification, inspection instructions, inspection schedules, feedback of measurement data to the original drawings. The CMM inspection plan is added to the CAD drawing and the combined specification is converted into the Initial Graphics Exchange Specification file format (IGES) (National Bureau of Standards 1988, Standards Association of Australia 1989). The measurement results from the inspection on the CMM are combined with the nominal dimensions from the initial input file into an IGES format type product inspection file. This file could then be transferred back to the CAD system and displayed in the form of the original CAD drawing.

As the IGES format standard was originally conceived primarily to facilitate the transfer of geometrical data between dissimilar CAD systems, it is not best suited to the bi-directional communication of inspection programs and measurement data between CAD systems and CMMs. The Dimensional Measuring Interface Specification (DMIS) (Aubin 1987, ANSI/CAM-I 1990, Fan et al 1992), funded by CAM-I and developed by the Illinois Institute of Technology Research Institute in 1986 was standardised by the American National Standards Institute in 1990. This standard was created specifically to provide a neutral format for the bi-directional communication of inspection data between CAD/CAM computer systems and computerised inspection equipment. DMIS supports three types of interface for the linking of CAD systems to CMMs, namely one-to-one, one-to-many, and many-to-many. A more detailed review of the data exchange standards is reported in chapter 3.5.3.1.
4.4.4 Inspection Integration into Automated Manufacturing

Over the past decade major contributions to the research into automated inspection within manufacturing industry has been achieved in the inspection of dies and moulds possessing sculptured surfaces. The drive for this area of research was spurned by the potential time and cost savings that could be achieved through the automation of what was traditionally a complicated, time consuming manual activity (Pahk et al 1993, 1995, Yau and Menq 1993). As product quality is becoming a fundamental characteristic in modern manufacturing establishments, automated inspection and information feedback is considered an essential aid in order to maintain a competitive edge within a global market. Efficiency of an automated manufacturing system has been regarded as dependent on both the quality and availability of feedback information regarding part status (Raja and Sheth 1988). An effective inspection system must acquire the information as soon after the manufacturing process as possible. Raja proposed an Integrated Metrology System (IMS) that utilised a CAD database interfaced to various special purpose metrology instruments. The facility was capable of not only dimensional inspection but was also employed to check out of roundness and surface finish.

A similar concept to the Raja’s integrated metrology system, in the sense that inspection could involve a number of completely different measuring instruments, was the strategic design driven inspection system developed by Galm and Merat (1988). This approach involves the use of fast sensors such as vision techniques for the initial inspection. Slow sensors such as CMMs were utilised to obtain inspection data of greater accuracy should any features require an inspection precision in excess of that possible by the fast sensors. The results obtained were transferred in IGES data format and subsequently analysed against the original CAD model.

A closed loop inspection system integrated into an automated machining cell was reported by Van den Berg which comprised of a CAD/CAM system, a CMM, machining centre, and a cell controller which were interfaced by a DECnet communications network (Van den Berg 1987). The system proposed can be applied to inspect manufactured components, consisting of sculptured surfaces, produced on the cell either by in-process inspection on the machining centre or upon completion on the CMM. The CAD/CAM system received the measured data for analysis, which was acquired from a CMM or a machining centre. The analysis was achieved in two ways: a) tolerance analysis which determines whether the component is accepted or rejected;
and b) manufacturing analysis that attempts to establish the manufacturing causes and proposes possible corrective action should discrepancies arise. The results of the analysis is communicated to the cell controller via the network.

A PC-based in-process inspection system, that incorporates adaptive feedback data, was developed at the National Taiwan University (Fan et al 1992). This approach involved the use of in-process inspection of the features on a component on completion of machining to a predetermined oversize condition. The results from the inspection were used to establish the actual required depth of finishing cut for each feature on the component thus eliminating the production of scrap due to overcutting conditions on the finish machining.

4.4.5 Automated Inspection Environments
Although the inspection activity itself is easily automated with the introduction of CMMs. The information support required to achieve a completely automated, flexible inspection facility that allows timely access to the appropriate, accurate data associated with other computer aided disciplines is a bewildering task. This issue has been the subject of many current research projects, as outlined below. The results of such research has been the creation of a number of computational and information environments to support the automated inspection of manufactured components and include:

i) an intelligent automated dimensional inspection environment (Menq et al 1989,1992, Yau and Menq 1992, Ge et al 1992);

ii) the integration of co-ordinate measuring machines with a design and manufacturing environment (Medland et al 1993);

iii) an architecture for integrated automated quality control (Reimann and Sarkis 1993, 1994).

The structure of an intelligent automated dimensional inspection environment for manufactured components incorporating a CMM to inspect free-form surfaces has been identified and established by Menq et al (Menq et al 1989,1992, Yau and Menq 1992, Ge et al 1992). Within the environment three levels of automation were addressed:
i) facility automation;
ii) information automation; and
iii) decision automation.

At the facility level, the CMM functionality requirements needed to inspect the component were examined. At the information level, a CAD-directed inspection system has been employed that consists of a CAD/CMM inspection planning module, a CAD based localisation algorithm, and comparative analysis module. At the decision level, artificial intelligence techniques were employed to automate the decision making aspect of inspection planning. This inspection environment has been implemented using IBM's CATIA CAD/CAM system and a CORDAX RS-30 CMM, the heart of which consists of five modules namely:

i) inspection specification module;
ii) automatic inspection planning module;
iii) CMM verification module;
iv) CMM execution module; and
v) comparative analysis module.

The environment is underpinned by a knowledge-based inspection planner that aids in process monitoring and assists in the decision making activity. The CAD/CMM interface is achieved through the co-operation of the CAD-directed inspection and the inspection planner.

Although the aforementioned inspection environment is primarily geared towards the specialised domain of die and mould manufacturing the IPSCIS research project conducted at Brunel University (Medland et al 1993) concentrated on an integration approach to CMM operation within an automated design and manufacturing environment. This research culminated into a feature-based approach to CMM operation that allows selective feature inspection and corrective action decisions to be executed within a harmonious computing environment. The environment facilitates the automatic creation of control programs, consisting of a small number of features, which can be utilised for in-process inspection as opposed to the large program needed to conduct final inspection. Thus the traditional functionality of the CMM of measuring
and reporting has been enhanced to a decision making role that is capable of assisting in manufacturing error and cause determination and process control.

An architecture for integrated automated quality control known as the Expert Programming System-One (EPS-1) was identified and created by Reimann and Sarkis (1993, 1994) at The University of Texas at Arlington. The structure of EPS-1 was based on the Computer Aided Manufacturing-International (CAM-I) Advanced Numerical Control (ANC) processor which was designed to automatically generate the necessary numerical control (NC) instruction sequence and associated support information required to manufacture prismatic components (Reimann and Sarkis 1994). The EPS-1 is a subset of the ANC architecture and differs in that inspection is not a metal removal process and therefore, certain characteristics of the ANC processor are not required for EPS-1. The EPS-1 processor interacts with external computerised systems which consist of:

i) **geometric modeller**: to identify and define the physical characteristics of the part. It can accommodate inspection type functions such as probe and part collision detection.

ii) **dimensioning and tolerance modeller**: to define tolerance nodes and assign these nodes to one or more geometric features and augment the geometric data of the part model.

iii) **applications interface specification (AIS)**: a neutral format specification which is similar to the IGES format which was an attempt to standardise the format of data exchange between any combination of application and geometric modeller and to allow access to the geometric modeller’s functionality.

iv) **dimensioning and tolerancing applications interface specification (DTAIS)**: which was similar to the AIS but allows both the exchange of D&T information and access to the D&T modeller’s functionality.

The DMIS neutral format specification was employed to provide the interface between EPS-1 and the dimensional measuring equipment (DME). The EPS-1 architecture has been decomposed into nine hierarchical modules or activities as listed below:
i) obtain operation plan;  
ii) task decomposition;  
iii) determine method and DME;  
iv) determine set-up;  
v) determine probe/holder;  
vi) detail/optimise operation plan;  
vii) generate/simulate probe path;  
viii) produce control information;  
ix) produce support information.

4.4.6 Inspection Based on CAD Models

A large proportion of research into the automation of the inspection activity has focused only on the automatic planning and production of inspection programs for CMMs (Raja and Sheth 1988, Corrigall 1990, Corrigall and Bell 1991, Reimann and Sarkis 1993, 1994, Spyridi and Requicha 1994). As previously mentioned in sections 4.4.3 and 4.4.5 there has also been a significant amount of research into utilising the CAD geometrical model to analyse the results acquired from the inspection of the component on a CMM.

Early research undertaken by Duffle (1988) involved the use of a tricubic solid cell database for representing not only the solid part geometry of sculptured surfaces but also the generation of machining motions and manufacturing process compensation. Shape error analysis was achieved through iterative searching the tricubic solid database to ascertain the points on the CAD model corresponding to each inspected data point. These points were superimposed, in the form of error vectors displaying both the error magnitude and direction.

One approach adopted by many researchers was to manipulate the inspection results, by using best fitting algorithms to assign features to raw point data, to produce a CAD model representation of the inspection results which was then compared both mathematically and visually with the original CAD model. This research direction has also been employed to produce the initial CAD representation from the inspection of a prototype or sample component which can be used as an electronic gauge, often referred to as a soft gauge (Requicha 1989), for future inspections. This by-product of automatic inspection research has received a great deal of attention over the past five years and is referred to as reverse engineering (Bidandra et al 1991, 1994, 1995, Gupta and Sagar 1993, Sobh et al 1994).

Galm and Merat's (1988) strategic design driven approach to component inspection involved the construction of both reference and object models. The reference model was deduced from an IGES generated file from the CAD system. The basic format of the model consisted of a complete, unambiguous surface adjacency graph
(SAG) that represents the objects surfaces and their relationships. The object model was also constructed in the form of a partial or SAG subgraph from data obtained from a fast sensor such as a vision system. The incomplete subgraph constructed from only a single visible view of the fast sensor was augmented by object manipulation or additional views from the sensor. The complete reference and object models are compared feature by feature. The tolerance specification of a particular feature in the reference model was examined and the conformance of the corresponding feature in the object model was checked. If the feature required measurement using an instrument of increased accuracy then that feature is added to a list for subsequent refinement on a slower sensor. The activity of the object model refinement was iterative using progressively more accurate measuring instruments until the inspection was completed to the desired accuracy.

About the same time, the intelligent CAD-CMM interface created by Cowling and Mullinuex (1989) utilised a three model concept. The model of the component on the CAD system, referred to as the original CAD model, was utilised in the inspection strategy decision process. A second model held at the CMM was termed the physical component model. The data points returned by the CMM via the interface were manipulated, with curve fitting, translation and rotation and end point determination algorithms, to construct a third model within the CAD system known as the reformed CAD Model. The differences between the original and reformed CAD models represent where manufacturing errors exist.

As previously mentioned, one variation into the production of a number of models within the CAD environment was identified by Farmer and Smith (1991) that involved the merging of the inspection results with the original CAD representation and displaying them in the form of measured dimensions with the corresponding nominal dimensions on the CAD model. Although the aforementioned system was implemented on the AutoCad drafting package, a similar method was adopted by Hassan et al (1992) which used the functionality of the RASOR constraint modeller. RASOR merges the measured data from the CMM with the CAD feature file that is subsequently employed to generate a feature by feature report. Typical attributes of the report include: feature name, type, nominal and inspected values, offset errors and the CMM software confidence limits. RASOR tried to identify causes of manufacturing errors (Rentoul et al 1994) by attempting to match the nominal and actual data through the least squares fitting technique in conjunction with a translation and a rotation. The results of the analysis
could be displayed on the RASOR user interface in colour coded form or transferred to the CAD system to be superimposed onto the original CAD model.

4.4.7 Emerging Research Issues in the Context of CAD-directed Inspection

It has been recognised by Raja et al that an efficient automated manufacturing system is dependent upon the availability and the accuracy of feedback information regarding part status (Raja and Sheth 1988). Computerised inspection systems, such as CMMs are regarded by many as the essential tool required to achieve optimal production efficiency. Early efforts by Hopp into CAD directed inspection involved the decomposition of the inspection activity into sub-tasks that form a hierarchy of control consisting of seven levels of control ranging from tolerance specifying to issuing of servo commands (Hopp 1984). With this system tolerances were required to be specified interactively by the user and the setting up of the CMM was achieved manually by the operator. The system is currently restricted to two-dimensional components and it is unclear as to how the measured point data is analysed and reported.

The EPS-1 approach reported by Reimann and Sarkis provided a limited capability of supporting three-dimensional components consisting of planes, cylinders, quadratics and cube type features and allowed CMM set-ups to be automatically determined (Reimann and Sarkis 1993, 1994). This approach decomposed the inspection task into nine activities which was based on an existing system developed for the automatic production of NC part programs known as the ANC processor. The EPS-1 was capable of producing DMIS code and support information for the CMM operator without the need for human interaction, however, the EPS-1 processor does not possess any inspection analysis capability and there is no mention of any strategy for the interpretation of measurement point data.

There have been two distinct research directions adopted by researchers in the field of CAD-directed inspection. The first direction adopted by Mullineux and Singh et al was to integrate the CAD and CMM facilities by providing a physical link that possessed local intelligence (Mullineux 1988, Singh et al 1990). Their research concerned itself with the provision of an intelligent interface to manipulate and process the inspection data in order to free the CAD system from the laborious and time consuming activity of managing large amounts of measurement data. However, this approach possesses one major drawback, if the intelligent interface does not employ the same feature fitting algorithms as the partner CAD system uses to create the initial model.
For example, in the case where high precision inspection is required, the differing algorithms can produce conflicting feature parameters for the same data points. This scenario could lead to an acceptable component being rejected or an inferior one being accepted. This problem is also very apparent with contemporary direct computer controlled CMMs that best fit features and analyse the results locally. The second method involves the transferring, via a standard neutral data format such as IGES or DMIS, of the raw inspection data for analysis in the CAD system. This ensures that the same algorithms are used not only for the creation of the original CAD model but are also used to analyse the measured point data. This approach does have the disadvantage of burdening the CAD system with measured point analysis computations.

4.4.8 Critique – Inspection Results Evaluation

The systems proposed by both Raja and Sheth's IMS project concept of interfacing a CAD database with a number of specialised metrology instruments and the Galm and Merat’s approach using fast sensors for initial inspection and slower sensors for the model refinement, require a major investment in metrology equipment (Raja and Sheth 1988, Galm and Merat 1988). These inspection applications are viable concepts for large manufacturing enterprises having stable product ranges, but would prove unsuitable for SMEs, having a limited investment budget, operating in an environment of dynamic customer demands.

As almost all CAD systems available, at present, do not support a tolerancing capability. The automated inspection systems described by Menq et al, Walker and Wallis require the addition of dimensional and geometric tolerances to be assigned to the CAD model in the form of attributes (Menq et al 1989, 1992, Yau and Menq 1992, Walker and Wallis 1992). Other researchers like Galm and Merat do not address the difficult problem of tolerance assignment and assume that tolerances have already been incorporated into the CAD representation prior to invoking the inspection application. However, the data representations of the CAD models are currently being enhanced to facilitate the capture of all geometrical and non-geometrical information relevant to the life cycle of the product to form a product model. A detailed description of product modelling and product data exchange specifications are discussed in chapter 3.5.

The emerging neutral data exchange formats are striving to eliminate the device dependency of computer aided engineering systems. However, these standards are at
present being continually updated and complete compatibility between different versions of the same standard is not guaranteed.

It has been noted from the literature discussed in this section that the information requirements to support the diverse hardware systems available today is an extremely complex issue. Therefore, the need for complete and accurate CAD/product models and consistent, globally accepted data exchange standards is of crucial importance for the comprehensive automation and integration of CAD, CAM and CAI systems.

4.5 Current State of Commercial Inspection Systems

As previously mentioned there has been a multitude of experimental systems conceived throughout the academic community which address the automation of the activities involved within CAI systems. However, the responsibility for the adoption of these experimental concepts and implementing them within commercially available CAI systems is borne by specialist software vendors and CMM manufacturers. This section provides a brief insight into off the shelf CAI software solutions that are currently available.

4.5.1 Automation Software's PC-DMIS

PC-DMIS® geometric measurement software is a PC based system specifically developed to be retrofitted on both manual and DCC CMMs. The system operates under the Windows environment and provides the seamless bi-directional exchange of information, via IGES and DMIS, between CMMs and CAD systems for on-line graphics-driven inspection programming and reverse engineering (Automation Software 1995, 1996a, 1996b). The powerful graphic user interface allows an interactive graphic representation of the actual component which allows the user to measure and program the inspection of parts by user interface directed interaction. Programs can be debugged via computer simulation of an animated probe. PC-DMIS provides a powerful analytical reporting capability that employs either textual or graphical style representation. Links are provided to allow interaction with other analytical and reporting packages such as Statistical Process Control (SPC) (Automation Software 1996c) systems.

4.5.2 International Metrology System's Virtual-DMIS

Virtual - DMIS® developed by International Metrology Systems (1997) provides a similar graphic user interface for the on-line creation of DMIS inspection programs and
the subsequent analysis of inspection results. Virtual – DMIS allows the import of CAD data in the form of either IGES, DXF, VDA, STEP or VRML file formats. Simulation of the probing path and CMM movements is achieved through the visualisation of the relative motions on a virtual CMM. Output from the system can be in the form of either: a direct read out, a spreadsheet type document comprising of feature nominals, actuals and their deviations or the resultant data can be exported to a SPC application.

4.5.3 LK’s Visual CMES

LK is a typical CMM manufacturer situated in the East Midlands of the UK, which provides as standard with all of their CMMs the Windows™ based Visual CMES software (LK 1998a). Visual CMES allows easy on/off line programming and inspection of geometric features. All programs generated with this system are produce in LK’s own inspection programming language CMES. This system utilises icons rather than text to direct the user through the programming activity. Other software option available from LK include: Digigraph scanning software, surface software and DMIS software (LK 1998b, 1998c, 1998d).

4.5.4 Pathtrace’s EdgeCAM – CMM/DMIS

EdgeCAM CMM™/DMISTM (Pathtrace Engineering Systems Ltd 1995, Kaimet Systems 1997) forms part of Pathtrace’s EdgeCAM PC based software suite that also contains CAM, milling, turning, wire erosion and general modules. EdgeCAM CMM™/DMISTM is an off-line inspection planning system for DCC CMMs. A full range of CAD tools are available for the construction and editing of 2D and 3D part models. Models from third party CAD systems can be imported via the IGES, VDA, IGDS, Microstation and DXF file formats. Output of the CMM™/DMISTM module is a verified DMIS input file that can be executed automatically on a DCC CMM.

4.5.5 Tecnomatix Technologies Inc. - Valisys

Valisys™ produced by Tecnomatix Technologies Inc. (1998) is probably the most comprehensive commercially available CAI systems to date. Valisys™ primary function is to define, predict, measure and analyse manufacturing tolerances throughout the industrial process. Valisys™ consists of seven individual modules that can be fully integrated with Unigraphics™, Catia™ and CADDSS™ CAD systems on a UNIX™
based platform. These seven modules comprise of the following (Tecnomatix Technologies Inc. 1998):

i) **Valisys/Design** provides the vehicle for the addition of manufacturing tolerances in the form of a GD&T specification to the CAD Model. This is achieved through the application of electronic tolerance zones called softgauges that are stored within the master CAD model;

ii) **Valisys/Assembly** is employed to perform 3D tolerance stack-up analysis and predicts the variation that will occur in an assembly and identifies the contributing tolerances that cause the problem;

iii) **Valisys/Reverse** allows the creation of 3D CAD models physical prototype components or master tooling;

iv) **Valisys/Programming** generates off-line machine independent inspection programs for CMMs and NC machine tools. Flexible programming is supported through the ability to produce both DMIS programs or direct control through a machine tool interface;

v) **Valisys/CMM** is employed to produce off-line machine independent collision free DMIS programs for CMMs;

vi) **Valisys/Inspection** provides a tool for inspecting and analysing components in a shop floor environment. This module supports over 30 types of CMMs and NC machine tools and allows on-line modification of inspection programs during execution;

vii) **Valisys/Analyse** fulfils the task of comparing the actual measured data with the nominal specification contained within the master CAD model. This module conducts a consistent analysis on the data regardless of the measuring device employed. The actuals and the nominals can be graphically displayed simultaneously. The actuals can be floated over the master CAD representation to ascertain the best possible fit.

A machine tool interface (MTI) provides the means for handling all on-line communication between the workstation and the inspection device.
4.5.6 Critique - Current State of Commercial Inspection Systems

As CMM and inspection software vendors deal with the latter product life-cycle activities, the only input from other manufacturing applications is a geometrical representation of the component in the form of a neutral format file i.e. IGES, VDA, or DXF files. This type of file is limited to transferring geometrical data, and as the standard was originally created to allow transfer of geometrical data between CAD systems, it does not support inter-application communication of manufacturing data namely, feature definitions and tolerance information. Feature and tolerance information must be interactively assigned to the geometric entities at the onset of the inspection planning activity.

CMM manufacturers provide primarily software systems to support the control and analysis of the inspection process on their own products. As a consequence of this focused objective, CMM manufacturers have little or no interest in providing interfaces to other software vendor products, for example CAM systems or diagnostic expert systems. This provides an impenetrable barrier that must be overcome if the integration of all manufacturing phases of the product life-cycle is to be achieved.

The Valisys™ (Tecnomatix Technologies Inc. 1998), system epitomises the same short-comings as the research contributions reported earlier in this chapter, as it is extremely complicated and very expensive in terms of both cost and the user expertise requirement. Therefore, its application is restricted to installations within large and deterministic enterprises, such as the automotive (Lederer 1996), aerospace and military (Miller 1988) industries.

EdgeCMM™/DMISTM, on the other hand being a module of the Pathtrace’s EdgeCAM suite of software products, has its functionality limited to the inspection planning and code generation activity only. There is not, as yet, a results analysis module available to interpret the inspection results once obtained.

It can clearly be noted that these systems act as stand-alone systems the are inherently closed in nature and do not possess the flexibility to be customised to the requirements of the end user.
Chapter 5

MANUFACTURING FAULT DIAGNOSIS APPROACHES

5.1 Introduction
As previously mentioned it has been recognised that inspection and measurement of manufactured parts is undertaken for two main reasons (ElMaragh and ElMaragh 1994):

i) Product control - to verify conformance of the component to the design intent;
ii) Process control - to provide feedback to achieve tighter control of previous manufacturing processes.

As a consequence of increased customer pressure for improved quality at reduced cost, an ideal closed loop manufacturing system requires to encompass both the product and process control inspection scenarios and must incorporate the following (Anjanappa et al 1990, 1996):

i) Part measurement;
ii) Determination of geometric errors through comparative tolerance analysis;
iii) Determine the most probable production cause for geometric errors; and
iv) Recommend corrective actions to eradicate the problem.

The research reported in chapter 4 has purely been concerned with the activities of part measurement and geometric error determination without addressing the issues of production error evaluation and corrective action initiation required to assure manufacturing recovery.

This chapter is concerned with the reporting of the techniques and experimental prototype systems that can be employed in conducting manufacturing error diagnosis and covers the following sections:
Traditional Troubleshooting Concepts

The conventional approach to troubleshooting of industrial problems, in which a solution is not immediately evident, involves the gathering of relevant information, the formulation of hypotheses on possible causes, and identification of potential solutions and then applying each solution in turn until the problem is solved (Gillespie 1988). Although this methodology does not require any special effort to solve simple problems, it is iterative in nature and does require the troubleshooter to possess prior knowledge of the specific problem space.

The are a number of formal approaches that have emerged from the intense effort being exerted for continuous improvement which can be employed to assist the troubleshooter to solve complex industrial problems first time, with the minimum of expense in both time and cost (Gillespie 1988, Juran and Gryna 1988a):

i) Checklists;  vi) Quality Circle method;
ii) What Changed approach;  vii) Relevance Trees;
iii) Morphological approach;  viii) Statistical Process Control;
iv) Brainstorming;  ix) Expert Systems;
v) Weighted-Factor Analysis;  x) Neural Networks.

The first seven approaches represent simple manually related techniques that have been employed to assist in decision support for many years. The remainder constitutes more advanced structured approaches to problem solving. Since these structured methods are more complex in nature they tend to rely more on computer assistance for their execution. A brief overview of the basic troubleshooting techniques followed by a detailed appraisal of the structured methodologies forms the scope of the remaining part of this section.

5.2.1 Checklist Approach

The checklist approach ensures that all the issues pertaining to the problem in question have been considered. This method guides the troubleshooter through a predefined
procedure and proves to be very effective when there are many factors and issues involved. This approach can be employed when the troubleshooter’s knowledge of the problem space is limited. Although checklists do not directly solve problems they do imply possible solutions.

5.2.2 What Changed Approach

One essential issue with any troubleshooting task is the emphasis on what has changed as a consequence of the problem. By concentrating on these changes provides a clue to the causes that induced the changes. This directed analysis saves considerable time on both the conventional and checklist troubleshooting methods. If concentrating on what changed does not resolve the problem then an alternative approach must be utilised.

5.2.3 Morphological Approach

This approach is used to formulate solutions to problems that occur through the interaction of a number of variables. This methodology involves the identification of all possible pairs of key operational parameters that can influence the process. This list is expressed in the form of a square table with the parameter list forming the header for both the columns and rows. A grid is constructed that represents all the other parameters that a particular variable can react with. Those combinations of variables that can cause the problem under review are highlighted by darkening the cell in the grid that represents the variable interactions. As each of these interactions is reviewed to determine potential problem areas, a list of how these parameter interactions can affect the problem is formulated. This technique is best employed when a complete list of the issues is required. Morphological analysis often produces better results within a team or group situation as a more critical review of the variable combinations can be explored.

5.2.4 Brainstorming

The objective of brainstorming is to develop an exhaustive list of ideas and theories about a subject (Winchell 1991). Brainstorming is generally employed when normal ideas fail to resolve the problem. It normally employs four to eight people that indulge in an unrestricted flow of ideas within a certain period of time. Although this technique is applicable to assist in solving a variety of problems in any subject, in troubleshooting
it helps generate a rapid, unrestricted collection of ideas and as many theories as possible (Gillespie 1988, Juran and Gryna 1988a).

5.2.5 *Weighted Factor Analysis*

This approach can be applied once a number of possible causes or solutions have been established. A probability or weighting factor is assigned to the cause or solution, which is dependent on its perceived importance. The cause or solution with the highest score is then pursued. Although this technique is sophisticated it only provides a quick fire answer as the assignment of weighting factors is very subjective (Gillespie 1988).

5.2.6 *Quality Circles*

On of the major characteristics Japanese company wide quality control is the Quality Control Circle Movement started in 1962 and forms a major contributor to the Japanese philosophy of Continuous Improvement. The nature and role of quality circles consists of a voluntary group of five to ten workers from the same work area, who meet on a regular basis. These meetings are co-ordinated by a Forman, assistant Forman, or one of the workers. The members of the circle employ statistical quality control and related methods to achieve significant results in quality improvement, cost reduction, productivity and safety. The six quality control tools at the disposal of the members in pursuing continuous improvement include (Gillespie 1988, Juran and Gryna 1988a, Sytsma and Manley 1998):

i) Pareto charts;

ii) Cause and effect diagrams (also known as Ishikawa or fishbone diagrams);

iii) Check sheets;

iv) Histograms;

v) Scatter diagrams; and

vi) Shewhart’s control charts and graphs.

All members of the circle are continually engaged in self and mutual development, control and improvement whenever possible.
5.2.7 Relevance Trees

Relevance trees provide a simple graphical representation of the relationships, using the same information as that presented in the morphological and checklist methodologies. The graphical representation provides a quicker source of reference than that of a compiled list of relationships.

5.2.8 Critique- Traditional Troubleshooting Concepts

The traditional approaches for troubleshooting of process problems rely heavily on human interaction in the form of a troubleshooting team. These approaches require a considerable amount of time in order to gather, collate and analyse the process information whilst not being able to guarantee that the analysis is meaningful. None of these manual techniques explicitly provides a solution to a problem, but they aid in directing the problem solving activity in an orderly, iterative manner. These methods need not be employed in isolation and can be applied to solve problems in any problem domain, however, these methods, with the exception of check lists, do require a considerable amount of localised expertise on behalf of the troubleshooting team. This localised expertise can produce subjective and inconsistent solutions as it relies heavily on the recollection skills of the experts involve in the problem solving process.

5.3 Structured Troubleshooting Approaches

There are a number of troubleshooting tools available to the practitioner that follow a more structured and analytical approach to troubleshooting than the methodologies previously mentioned. These structured methodologies include: statistically based process control, expert systems and neural networks.

5.3.1 Statistical Process Control (SPC)

Shewhart's conceived the philosophy statistical quality control (SQC) at the Bell Laboratories in the latter half of the 1920s and can be defined as (Juran and Gryna 1988b):

"The application of statistical techniques for measuring and improving the quality of processes. SQC encompasses SPC, diagnostic tools, sampling plans and other statistical techniques."

SPC, which forms a significant element of SQC, is a method for defining and controlling process variability. All processes suffer from at least one of two forms of
process variability. Natural or assignable process variability, also commonly known as system variability, corresponds to the inherent fluctuation of the process over time. Special cause variations constitute variations caused by some problem or the occurrence of an abnormal disturbance (Sytsma and Manley 1998). The SPC approach adopts a methodology for the charting of variables or attributes samples (Juran and Gryna 1988b) of the process in order to determine when the process being monitored is out-of-control, viz. the process has encountered a special cause variation. In the context of SPC, variables represent continuous values that are positioned within a certain range whereas attributes relate to whether values are good or bad. The SPC tools available for process analysis includes: histograms, check/tally charts, Pareto analysis, cause and effect diagrams, scatter diagrams, process flow charts and control charts. Control charts can be further sub-divided into two categories depending on the data type being monitored is summarised in table 5.1.

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Chart Type</th>
<th>Value Charted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Variables</td>
<td>$X$ bar and $R$ chart</td>
<td>Sample averages and ranges</td>
</tr>
<tr>
<td></td>
<td>$X$ bar and $S$ chart</td>
<td>Sample averages and standard deviations</td>
</tr>
<tr>
<td></td>
<td>$X$ and moving $R$ chart</td>
<td>Individual observations and ranges</td>
</tr>
<tr>
<td></td>
<td>Median and $R$ chart</td>
<td>Sample medians and ranges</td>
</tr>
<tr>
<td>Attributes</td>
<td>$p$ chart</td>
<td>Proportion of units defective per sample</td>
</tr>
<tr>
<td></td>
<td>np chart</td>
<td>Number of units defective per sample</td>
</tr>
<tr>
<td></td>
<td>c chart</td>
<td>Number of defects per inspection unit</td>
</tr>
<tr>
<td></td>
<td>u chart</td>
<td>Av. Number of defects per production unit</td>
</tr>
</tbody>
</table>

Table 5.1 Commonly Used SPC Control Charts 1 (Gillespie 1988)

It must be noted that SPC does not possess the capability to solve problems directly, however, it organises the data about a process being investigated into a format that can assist in the identification of trends through the application of time based control charting. Once an out of control condition has been identified the root cause of the problem is determined and an appropriate corrective strategy employed. This subsequent investigation and correction usually involves a team effort and employs any number of the Total Quality Management (TQM) continuous improvement tools previously mentioned in section 5.2. The SPC activity can be decomposed into three stages:
i) An examination of the state of control of the process to ascertain the assignable and special case causes of process variation. Once these have been removed, the process is said to be in statistical control;

ii) Process capability study of process to ascertain whether the remaining variations from (i) are acceptable;

iii) Process control using control charts that employ a traffic light warning system through the application of warning and action control limits that notify machine operators real time information regarding the process and its current state of control.

A comprehensive outline of the SPC charting techniques, their advantages, disadvantages and their application areas is discussed in Juran’s Quality Control Handbook (Juran and Gryna 1988b) and recommends that for machine dominant processes variations, that $X$ bar and Range, $X$ bar and standard deviation type control charts provide the best results. The effective exploitation of SPC techniques can provide an improvement in productivity by preventing the production of defect products and avoiding the interruption to production through rework, whilst reducing quality costs in terms of reduced machine failure and component renovation costs.

Jennings and Drake (1997) have applied SQC charts for the condition monitoring of a Wadkin V4-6 vertical machining centre. Their approach utilised a novel one-variable, two-variable and three-variable control charts and the method of normalisation to compensate for parameter inter-dependence. A normalisation chart defines the normal relationships between the performance parameters. The difference between the current measurement and the corresponding value from the normalisation chart is employed to calculate the residual. These residual values are then plotted on control charts to monitor the condition of the machine tool. This approach has been applied to the monitoring of the work done, normalised control cabinet temperature and pressure, coolant flow and motor power of the coolant systems of the machine tool on one-, two- and three-variable type control charts respectively.

5.3.2 Expert System Paradigm

Expert systems are a branch of artificial intelligence (AI) that aims to attempt to emulate a human expert’s reasoning when applied to a specific problem. In this context
someone is considered an expert about a problem if he or she possesses specialised knowledge regarding the problem area. The specific, or domain knowledge as it is also known, is captured within his or her long-term memory.

When applied to problem solving the human expert first obtains the facts regarding the current state of the activity that is experiencing the problem. These facts are subsequently stored within his or her short-term memory. The expert then reasons about the problem by applying the knowledge contained in his or her long-term memory with the facts contained within his or her short-term memory. During this reasoning process the expert can infer additional facts that are stored within the short-term memory and eventually arrives at a conclusion or solution to the problem. (Durkin 1994a). Expert systems try to solve a problem by employing a similar method to that used by a human expert and as such consist of three elements:

i) Knowledge base module is analogous to the long-term memory of the human expert and is used to capture the expert’s domain knowledge. This domain knowledge is usually expressed in the form of IF and THEN types of rules. The IF or antecedent portion of the rule represents the condition whilst the THEN or consequent part represents the action to be initiated if the condition is satisfied;

ii) Working Memory is analogous to the short-term memory of the human expert and contains the facts about the problem. The facts are only stored for the duration of the consultation;

iii) Inference Engine emulates the process of human reasoning by matching the facts contained within the working memory with the domain knowledge captured in the knowledge base in order to postulate a conclusion to the problem.

A trademark of an expert system is the ability explain its reasoning to the user through an explanation facility, which can also assist the developer to resolve errors in the systems knowledge. Interaction between the expert system and the user is conducted through a user interface using a natural language style of communication and follows closely the human dialogue. The interaction and communication between the expert or knowledge engineer and the knowledge based is conducted through a knowledge acquisition sub-system.
There are a number of different expert system configurations that can be employed to assist in problem solving, the choice of which will be heavily dependent on the problem domain and the user requirements. These configurations include:

i) **Induction-based expert systems** use examples in the form of a table of attributes and values to represent the expert’s knowledge. An induction algorithm is employed to derive a decision tree from the table of attributes (Harmon and Sawyer 1990);

ii) **Rule-based expert systems** use facts to represent the problem, rules to represent the expert’s knowledge and employs forward or backward chaining to infer new facts or conclusions (Harmon and Sawyer 1990). Uncertainty within rule-based expert systems can be addressed with the addition of a certainty factor to the rules, as employed in the MYCIN medical expert system, or with the applications of fuzzy sets or fuzzy logic (Kandel 1992, Durkin 1994a);

iii) **Frame-based expert systems** possess functionality borrowed from the object-oriented programming paradigm and applies it the rule-based expert systems. The functionality includes the concept of objects, inheritance and message passing (Harmon and Sawyer 1990, Durkin 1994a).

The most popular choice of configuration adopted by knowledge engineers to date for building expert systems is the rule-based approach. The popularity has grown out of the large number of successful rule based systems built (Durkin 1993, 1994b) and the availability of expert system development software. These development software systems are collectively known as expert system shells and include: CLIPS (Giarratano and Riley 1998, Giarratano 1998), JESS, LEVEL5 OBJECT (Anon 98), KEE, KLUE (Karel and Kenner 1988), GURU and VP EXPERT, to name but a few (Durkin 1994c, Anon 1999).

5.3.2.1 Expert System Advantages within a Diagnostic Context

The application of expert systems within the field of diagnosis and troubleshooting has the ability to exploit any number of attractive features (Giarratano and Riley 1998):
i) Increased availability;
i) Increased availability;
ii) Reduced cost;
iii) Reduced danger;
iv) Multiple Expertise;
v) Permanence;
vi) Increased reliability;
vii) Explanation;
viii) Fast Response;
ix) Consistent, unemotional and complete response at all times;

As the knowledge is explicitly captured within an expert system, it can be examined and re-examined for correctness, consistency and completeness which improves the quality of the knowledge.

5.3.2.2 The Representation of Knowledge

Knowledge representation is of major importance in expert systems for two reasons. Firstly, expert systems are designed to operate on a certain type of knowledge representation such as rule or logic and secondly, the way in which an expert systems represents the knowledge affect the development, efficiency, speed and maintenance of the systems (Giarratano and Riley 1998). Knowledge as defined by Durkin (1994a) is an understanding of a subject area whilst he also defines knowledge representation as:

"The method used to encode knowledge, regarding a well focused subject area or domain, in an expert system's knowledge base."

The study of the philosophy of knowledge, known as epistemology, is required into the way in which humans obtain and apply knowledge to solve problems before knowledge representation can be addressed. Researchers in AI have applied the results of studies conducted by psychologists, on the cognitive abilities of humans in solving problems, in order to develop techniques to represent this knowledge in a computer form (Durkin 1994a).

Expert systems knowledge can be categorised into two classes namely shallow knowledge and deep knowledge. Shallow or heuristic knowledge represents knowledge based on judgement or experience rather than form first principles whereas, deep knowledge represents knowledge derived from either first principles, physical laws of the domain, or from knowledge that represents the process structure and behaviour
These basic categories can be further subdivided into specific types of knowledge (Giarratano and Riley 1998, Durkin 1994a):

i) **Procedural knowledge** describes how a problem should be solved;

ii) **Declarative knowledge** describes what is known about a problem;

iii) **Meta-knowledge** describes knowledge about knowledge. This knowledge is used to choose appropriate knowledge to solve a problem;

iv) **Heuristic Knowledge** describes a rule-of-thumb that guides the reasoning process and represents the past experience of the expert and as such is often referred to as shallow knowledge;

v) **Structural Knowledge** describes knowledge regarding structures in the form of concepts, sub-concepts and objects that comprise the expert’s overall mental model of the problem.

Not only has there been an overwhelming amount of research into what types of knowledge are employed by humans to solve problems, but also there has been an equivalent amount of interest from researchers into AI into the computerised representation of these types of knowledge. The most commonly used representation techniques include (Marshall 1990, Durkin 1994a, Giarratano and Riley 1998):

i) **Object-attribute-value triplets** \((O-A-V)\) are facts and as such constitutes declarative knowledge that is employed to assert a particular property to an object.

ii) **Rules.** As previously mentioned, user or system supplied facts allow the expert systems to understand the current state of the process. However, additional knowledge is required to analyse the facts in order to solve a problem. This additional knowledge is represented in a rule structure, where a rule is a form of procedural knowledge and associated facts to a given action. This action may be the assertion of new facts or perform a certain procedure;

iii) **Semantic networks** provided an early attempt to represent knowledge in computer form. It provides a hierarchical graphical view of a problem’s relevant objects, properties and relationships. Objects are represented by nodes whilst arcs represent the relationships between objects. Labels on the nodes and arcs describe the objects represented and their relationships respectively;
Frames, according to Durkin (1994a), are data structures for representing a concept or object. Bartlett in 1932 first coined the term schema to represent a structure that contains typical knowledge about some concept or object which was based on his study of how humans store experience in their long-term memory and adapt this knowledge for current situations. This schema, referred to as a frame by Minsky in 1975, has the capability of storing both declarative and procedural knowledge. Contemporary frame-based systems now incorporate most of the features found in the object-oriented programming paradigm to such a degree that the terms frame-based systems and object-oriented systems are interchangeable with no loss of definition.

Logic represents the oldest form of knowledge representation for use with computers and is the study of exact reasoning. A number of logic representations have been studied and proposed such as propositional logic and predicate logic (Giarratano and Riley 1998), the latter of which form the basis of PROLOG. Propositional logic represents and reasons with statements that are true or false. Logical operators such as AND, OR, NOT, IMPLIES and EQUIVALENCE allows reasoning with various rule structures. Predicate logic is a refinement of propositional logic that permits a finer representation of the knowledge to describe the relationships of the knowledge (Durkin 1994a). Logic is also of fundamental importance to expert systems that the inference engine reasons from facts to conclusions. Logic programming and expert systems can both be described by the term Automated Reasoning Systems.

5.3.2.3 Knowledge Acquisition and the Expert System's Development Cycle
The six stages of expert systems development is illustrated in figure 5.1 and is initiated with an assessment in the form of a requirements analysis of the problem to be solved (Lee 1990, Durkin 1994a). This requirements analysis phase of development aims to provide sufficient information regarding the problem domain so that initial development decisions can be made. The process would include interactive interviews with the project leader, domain expert or experts, and the end user. The interviews would be directed to answer typical questions such as (Lee 1990):
i) What is the system required to do?
ii) What knowledge sources are available?
iii) What are the end user requirements?
iv) What is the scope of the desired system?

The next phase of development is the knowledge acquisition, the primary objective of which is to gather and structure the specific body of knowledge regarding a problem’s domain so it can be encoded for use in an expert system. The knowledge acquisition stage of development is the most crucial in the development process as the effectiveness and accuracy of the resultant expert system is purely dependent on the quality of the knowledge held within its knowledge base. This knowledge can be sourced from documentation such as books, manuals, standards, regulations, reports or databases although in most cases the most predominant source of knowledge regarding a problem domain is obtained from a human expert. The extracting of knowledge from a human expert has been distinguished from the other forms of knowledge acquisition and is known as knowledge elicitation. The elicitation process can involve lengthy and
tedious interactive interview sessions or by the case study method which involves studying by observation of how the domain expert solves real problems. After accomplishing the initial collection of knowledge the system's developer must interpret, analyse and design and then code information into the expert systems. The developer must then test the system, analyse and then utilise the results to direct further knowledge acquisition sessions. This process is iterative in the sense that once the embryonic system has been developed and tested it proceeds into the maintenance phase. This phase involves the updating of the systems knowledge based with new knowledge acquired through new experiences and with the addition of new technologies to the system being analysed. Therefore, the process of knowledge acquisition continues through the complete life cycle of the expert systems.

5.3.3 Inference Strategies and Methods

Reasoning is the process by which humans combine facts with their understanding of a problem domain to derive logical conclusions. Humans reason in any one of a number of ways (Durkin 1994a, Giarratano and Riley 1998):

i) Deductive reasoning deduces new facts from logically related known facts;

ii) Inductive reasoning arrives at a general conclusion from a limited set of facts by a process of generalisation;

iii) Heuristic reasoning uses rule-of-thumb or experience to attain a conclusion;

iv) Abductive reasoning is similar to deductive reasoning but allows plausible conclusions to be inferred;

v) Analogical reasoning uses a mental model of some concept based on personal experience of a situation or an object. This model is compared with the current situation or object under analysis and is refined by addressing any specific differences to create a new model;

vi) Common-sense reasoning is not an exact logic as it relies totally upon personal experience or good judgement of the situation or object to draw conclusions;

vii) Non-monotonic reasoning deals with facts that do not remain static during the course of the consultation.
The inference strategies and methods adopted by expert systems try to emulate the reasoning capability adopted by humans to solve problems. Durkin (1994a) defines inference as:

"The process by which an expert system derives new information from known information".

The inference process is conducted within an expert system by the inference engine module. It combines the facts relating to the problem domain held within the working or short-term memory with the knowledge captured within the long-term memory. New information is inferred as a consequence of this process and held within the working memory, which in turn can subsequently influence the outcome of further inferences. There are a number of inference strategies that can be employed within expert systems to emulate reasoning:

i) Modus ponens; iii) Forward chaining;
ii) Resolution; iv) Backward chaining.

A modus ponens inference method is based on propositional logic and works with truth statements to infer new facts. Its rule of logic asserts that if A is true, and A implies that B is also true, then it can be assumed that B is also true. The method of inference is also known by a variety of other names: direct reasoning, law of detachment, and assuming the antecedent (Giarratano and Riley 1998).

A resolution inference strategy is employed in logical systems to determine the truth of an assertion in order to attempt to prove some goal. The resolution style of inference is confusing as it does not distinguish between goals, premises, or rules and therefore its loses sight of what it is trying to prove. Non-resolution or the natural deduction technique addresses this problem by attempting to prove some statement in a goal-oriented manner and employs the backward chaining rule of inference to accomplish this.

A forward chaining inference strategy begins with a set of known facts, new facts are generated using rules whose premises match the known facts. It continues this process until either a goal state is reached or until there are no further rules that can be satisfied by the known or derived facts (Durkin 1994a). The form of inference is data-driven and is similar to the modus ponens approach mentioned earlier. The order in which the rules fire when more than one rule is activated is controlled in forward
chaining inference by conflict resolution techniques (Durkin 1994a, Giarratano and Riley 1998, Giarratano 1998)

The backward chaining inference strategy attempts to prove a hypothesis or goal by gathering supporting information. This strategy begins with a goal to prove. The working memory is checked to ascertain whether the goal has already been proven. If this is not the case the systems searches its rule base looking for a rule or rules that contain the goal in their THEN part. This rule then becomes a goal rule. The systems then checks to see if the goal rule’s premises are listed in the working memory. If these premises are not present they become new goals, or sub-goals, to prove. This process continues in a recursive manner until the system comes across a premise that is not supported by any rule. This premise is referred to as the primitive. When a primitive is encountered the systems asks the user to provide additional information regarding it. The system then uses the supplied information to help solve both the sub-goals and ultimately the original goal. Backward chaining approach to inference is similar to that of hypothesis testing activity employed in human problem solving (Durkin 1994a).

5.3.4 Expert Systems Applied in Other Problem Domains
Expert systems development software systems and shells provide a generic solution to problem solving that can be applied within any field of expertise. Durkin (1993, 1994a, 1994b) conducted an extensive survey covering over 2500 existing expert systems which are categorised into 23 application areas ranging from agriculture to transportation the distribution of which is illustrated in figure 5.2.

![Figure 5.2. Number of Developed Expert Systems over Various Application Domains (Durkin 1994a)](image-url)
Although the adoption of expert systems solutions has covered the entire spectrum of application areas, it can clearly be seen that the most prolific areas for expert system employment are the business, manufacturing and medicine application domains which amounts to a total of 36.6% of total expert systems usage.

Not only are the application areas for expert systems diverse but so are the range of activities within the application domains. These tasks can range from control, design, diagnosis, instruction, interpretation, monitoring, planning, prediction, selection and simulation. Diagnostic expert systems amounts to 30% of all the above tasks which is primarily due to the high proportion of expert systems developed for the medicine, engineering and manufacturing application domains (Durkin 1994a).

5.3.5 *Neural Networks*

This section is intended to give the reader a brief insight into the paradigm of neural computing and neural networks. Although the field of neural computing is a vast and embryonic research area, the author considers that an in-depth discussion into the theory of neural networks is out of the scope of this thesis. For additional information regarding the history and underlying theory involved in artificial neural systems (ANS) and neural networks refer to Müller *et al* (1995) and Haykin (1999).

The ANS or connectionism type of programming paradigms arose in the 1980s and is based on concepts derived from research into the nature of the brain’s processing capability (Müller *et al* 1995). This initiated research interest in to neural networks, the impetus of which stems from two objectives:

i) The desire to understand the principles on which the human brain operates;

ii) The wish to develop machines that are capably of performing, in parallel, complex tasks that cannot be achieved by the sequentially operating methods of adopted in traditional computing.

Researchers have identified the potential of neural networks as the front-end of expert systems that operate using large amounts of sensory input whilst providing a real-time response to problem solving. The real-time response benefit has been demonstrated through the use of the travelling salesman problem by reducing the processing time for a conventional mainframe CPU of 1 hour to a neural network’s processing time of 0.1 seconds when applied to a problem with only 30 cities
Giarratano and Riley 1998). This phenomenal reduction can be attributed to the parallel processing capabilities of neural networks avoids computational explosion experienced with traditional computers.

An ANS basically consists of an analogue computer that employs simple processing elements connected in a highly parallel manner. Boolean or algorithmic type computations are applied by the processing elements on the inputs to the neuron. It is the weights that are associated to the inputs that provide the information stored within the system to the neuron's processing element. Any number of neurons can be connected together to form a network that consists of an input, output and hidden layers of nodes.

Neural nets are not programmed in the traditional sense they employ a learning algorithm, such as counter-propagation or back-propagation, from a given set of inputs and the corresponding output. The net learns by automatically adjusting the weights in the network that connects the neurons. Although the training exercise can take a number of hours to days to complete, once the learning process is accomplished, the net can respond very quickly.

A neural network provides a good solution when there is considerable empirical data and no algorithm exists within conventional computing to provide the required speed and accuracy. The other benefits of neural nets compared to conventional computing pertain to data storage and include (Giarratano and Riley 1998):

i) **Data storage is fault tolerant.** Portions of the net can be removed with only a slight degradation of the quality of the data stored;

ii) **Degradation of the quality of data stored is graceful.** There is no catastrophic loss of information as the degradation of data quality is proportional to the amount of net removed;

iii) **Data is naturally stored in the form of associative memory.** Only partial data is required to recall the complete stored information unlike conventional memory that uses an address to access stored information;

iv) **Nets can extrapolate and interpolate from stored information.** After training a net is capable of extrapolating to suggest relationships on new data;

v) **Nets have plasticity.** Even if a number of neurons are removed the net can be retrained to its original skill level if enough neurons remain, which is a quality that is possessed by the human brain.
5.3.5 Critique - Structured Troubleshooting Approaches

The application of structured troubleshooting approaches to problem solving is an attempt to reduce the inconsistencies that can occur through the adoption of the more traditional manual approaches. Although the application of SPC techniques provides an indication of the onset of process variability through the time-based monitoring of process variables and attributes it cannot directly identify the cause or propose a solution to a problem. SPC does however, organise the evidence in a graphical format the can assist in identifying trends and imply where the expert should look for problem causes.

All of the manual troubleshooting approaches and SPC techniques require the expert to possess specialised domain expertise in order to interpret the facts to infer probable causes and possible solutions and are therefore inappropriate for application where this expertise is not available. Expert systems provide a method of representing the expert's domain knowledge in the form of a knowledge base. The acquisition and representation of this knowledge is of considerable importance to expert systems as it can adversely affect the efficiency and accuracy of the diagnostic process. It is for this reason the evolution of an expert system requires a full time development team and in almost all cases produces unique systems that can only be applied to the problem domain for which it was intended. As the expert system evolves and the size of the knowledge base increases the more complicated and time consuming the systems maintenance becomes. This complexity issue coupled with the uniqueness of developed expert systems are the reasons why there are no standard commercially available systems on the market. However, there are a multitude of expert system development shells that provide the developer with an empty expert system that only require the addition of process knowledge. The choice of which is wholly dependent on the application domain.

Neural networks are emerging as a means of analysing in parallel the input information of expert systems in order to reduce the response times that are currently being experienced with conventional computing. As this technology is still in its infancy it has not yet realised is full potential and is limited to diagnostic applications developed in research institutions.
5.4 Machine Tool Error Diagnosis

Machine tool error diagnosis is the troubleshooting activity of diagnosing either, faults due to manufacturing process or errors that occur due to the malfunction of some elements of the machine tool. The methods employed to achieve this can be classified into two distinct categories:

i) The condition monitoring of machine tools by using sensory data to attempt to ascertain the current condition of the machine tool;

ii) The inferring of machine tool errors through the inspection and manufacturing data analysis of components produced by a number of operations conducted on the machine tool.

This section introduces research into the classification of machine tool errors and outlines contemporary experimental research into diagnosing these manufacturing errors through both condition monitoring and manufacturing data analysis.

5.4.1 Manufacturing Fault Classification

In order to conduct a diagnostic troubleshooting exercise on a machine, one must first be aware of the possible machining errors that can occur during the machining process. A comprehensive study of the operating parameters and the machining errors for the whole spectrum of manufacturing processes was undertaken by Drozda and Wick (1983) and Gillespie (1998) on behalf of the Society of Manufacturing Engineers. The manufacturing processes explored includes:

i) Sawing;

ii) Broaching, planing, & shaping;

iii) Turning and boring;

iv) Drilling and reaming;

v) Milling;

vi) Grinding;

vii) Threading;

viii) Gear and spline production; and

ix) Non-traditional machining.

In general machining errors can be categorised into one of two classes: random or stochastic errors and systematic errors (Yandayan and Burdekin 1997a). Random or stochastic errors are the type that cannot be controlled by the operator and comprise the variations within the machine tool and the application variations introduced during use.
These errors can be attributed to a combination of the machine's structural integrity and condition and the errors due to the operator or control system. Systematic errors represent errors that cause a significant drift of measured results obtained from a number of workpieces over a period of time. Typical systematic errors include: thermal distortions of machine tools, tool wear, deflections of the machine/tooling/workpieces during machining, deflection of machine tool due to workpiece's weight and misalignment of the machine tool's axes.

Kramer and Nadanasundaram (1991) and Anjanappa et al (1990, 1996) have categorised the possible machining errors associated with the milling operation in particular. These categories include:

i) *Cutting tool errors*: incorrect tool size, incorrect tool type, runout, tool wear, Tool deflection due to improperly set speed, feed and depth of cut parameters;

ii) *Machine errors*: position errors, such as out-of-calibration, servo lag, X-Y positioning and squareness, thermal and stochastic errors;

iii) *Fixture/Workholding errors*: part/fixture location, fixture/machine location, part/machine location and chip control;

iv) *General Errors*: stock size, workpiece deflection due to clamping forces, vibration and chatter.

5.4.2 Condition Monitoring of Manufacturing Machine Tools

Condition monitoring can be defined as the real-time activity of observing sensory information to monitor either the machine tool condition or to monitor the machining process itself. Condition monitoring techniques can be applied to observe machine process parameters such as motor horsepower, cutting force, cutting temperature, vibration and acoustic emissions.

Vibration monitoring of tool failure proves to be the most popular method for the condition monitoring of cutting tool failure. El-Wardany et al (1995) employed vibration signature analysis techniques for the monitoring of tool failure in drilling operations on cast iron test pieces using a YAM 2½ axis CNC machining centre. Monitoring was achieved by detecting changes in the pattern of the vibration signals in the time domain during drill breakage of drills smaller than 3 mm in diameter.

Moore and Kiss (1995) uses vibration monitoring techniques for the detection of carbide insert fracture. Their methodology employed the amplitude probability density
function of the vibration signal to determine insert failure and involved recording the vibration signal, for subsequent analysis within a laboratory, of the middle third of the cutting cycle of a three carbide tipped face-mill on a mild steel workpiece. As a consequence of this their approach provides no real-time advantages. Although their approach was capable of determining the presence of an insert fracture, at present it is unable to quantitatively determine insert wear and therefore the onset of insert fracture.

The vibration signature of tool wear is also employed by Nicolescu and Bejhem (1995) for the on-line tool condition monitoring of turning operations, which employs statistical methods to compute the tool wear index which is used for monitoring tool life.

An alternative method for the diagnosis of tool wear is that proposed by Jemielniak et al (1998). Their approach employed cutting forces and acoustic emissions for tool wear diagnosis. These cutting parameters form the input to feed forward, back propagation (FFBP) neural network. They employ both conventional and unconventional training strategies and monitored the network’s response. Conventional training that relied on random initial weight values for the inputs led to over-training of the net and therefore a degradation of the net response. This response is greatly improved by introducing random distortions to the input weighting systems.

Konrad et al (1995) also employed cutting forces to determine tooling faults of carbide tipped cutters in milling. From cutting force measurements a force model is constructed that enables parameters to be estimated for each insert of the milling cutter. A classification algorithm analyses the pattern of the parameters in order to categorise insert condition into one of four classes: normal cutting, wear, breakage and radial insert initial displacement. This approach uses only cutting force in its parameter estimation with no consideration given for the influences caused by the parameters of cutting speed and feed per tooth.

An artificial intelligence system for estimating and in-process compensation of manufacturing process errors in CNC machining proposed by Zhou et al (1995) uses a neural network combined with a linguistic rule-based fuzzy controller. They employ a three stage method for compensating process errors caused by inherent geometric errors of the machine tool, process dependent errors and environment errors etc: (i) calibration stage, (ii) learning or training stage, (iii) real compensation stage.
The calibration stage involves the geometric inspection of a test piece on the machine tool after cutting and then removing the test piece and repeating the inspection on a CMM. A process error is computed that represents both the machining process error and the machine tool’s inherent error and forms the input to the fuzzy neural controller. The learning and training stage of the fuzzy neural network tries to reduce environmental errors by adjusting the rigid fuzzy memberships, which are used in the fuzzy control rule base. This is achieved by adjusting the weights of the process error and change of error that form the inputs to the fuzzy neural network. The final stage is the real compensation stage that involves the generation of a decision table of all the input variables from which a manufacturing process compensation value can be obtained.

An expert system for diagnosing faults in CNC machine tools is proposed by Bohez and Thieravarut (1997). The diagnostic model employed in this approach used a hybrid reasoning method that utilises both deep and shallow knowledge models. The shallow model represents the heuristic fault knowledge whilst the structure and behaviour of the CNC systems is represented within the deep model. The expert system utilises the maintenance manual procedures to diagnose controller malfunctions and relay ladder logic and electrical diagrams for the diagnosis of machine tool failures. The system employs the forward chaining inference principle and has been developed using the VP Expert systems shell and comprises 500 rules, 160 rules regarding controller troubleshooting and 340 rules for machine tool fault diagnosis, within its knowledge base.

As previously mentioned in section 5.3.1, Jennings and Drake (1997) devised a method of applying statistical process control variable type charts to continuously monitor the condition of a vertical milling machine tool and applied it to fault diagnosis of the machine tool’s coolant system.

The aforementioned review of condition monitoring techniques is by no means exhaustive (Martin 1994), however it is intended to provide the reader with an appreciation of the diverse approaches that can be adopted to provide real-time control of manufacturing processes and machine tools. Other condition monitoring examples that relate to the maintenance and process control of CNC machine tools and Flexible Manufacturing Systems (FMS) include: Puetz and Eichhorn (1987), Majstorovic and Milacic (1989), Lee (1995), Ye (1996).
5.4.3 Manufacturing Data Analysis of Production Errors

Manufacturing data analysis has been defined by Lee (1990) as the "analysis and feedback of manufacturing data" and is directed toward the determination of production errors, causes and the provision of corrective action feedback from geometric deviations obtained from the inspection of a component part.

An early attempt to address the problem of manufacturing data analysis was the closed loop inspection system for sculptured surfaces proposed by Van den Berg (1987). Van den Berg identified that manufacturing analysis consists of two stages:

i) The matching of the observed errors with the possible sources of error contained in a manufacturing process model; and

ii) Applying a corrective strategy to the manufacturing process.

Van den Berg’s theory acknowledged that the process of manufacturing data analysis for complex surfaces was a relatively simple task as the production of such surfaces involved only a few types of discrete cutter operations. According to Van den Berg the number of possible sources of manufacturing error increases factorially with the number of manufacturing processes between inspections, as in the case of prismatic components possessing many features. Van den Berg (1997) later elaborated and applied this initial proposal to the shape error compensation to tooling for both CNC milling and the formed tooling used in electro-chemical machining (ECM). The approach is employed for first-off component manufacturing, and involves design of the component, creation of a shaping plan, make sample part and the inspection of the part using a laser scanner. A tool path correction algorithm (TOPAC) produces a corrected shaping plan if shape errors are detected. This approach has been applied to the machining and shape error compensation of airfoil type components.

Pfeifer and Held (1989) proposed a backward chaining expert systems designed to diagnose the type, location, fault causes and recommended ways of eliminating them. The off-line prototype expert system relies on human interaction to establish potential fault causes, locations and remedies. This approach employs a static diagnostic decision tree to represent and assign geometric features to specific machine components responsible for producing them. Although this approach was proven with the application of the expert system to try and diagnose machine errors for a simple
threaded bolt consisting of four features, the prototype system's static knowledge tree is large and only covers one simple component and one machine.

Anjanappa et al (1990, 1996) applied a procedural rule-based approach in the development of their Computer-Aided Inspection Data Analyser (CAIDA) at the University of Maryland. This PC-based diagnosis approach attempts to determine milling operation errors from workpiece dimensional inaccuracies determined by the inspection of a component part on a CMM. The system is feature-based and supports hole, slot and pocket type features. The procedure adopted consists of two phases, namely: (i) individual error analysis and (ii) combined error analysis.

Individual error analysis involves the boundary size, slot and pocket edge analysis, X and Y hole position analysis and hole and diameter analysis of an individual feature and is capable of identifying cutting tool, machine tool, fixture set-up and stock boundary errors. Combined error analysis is an attempt to filter out any incorrect error assertions and comprises two analyses: combined tool error analysis and combine machine and fixture analysis. Although the system reported is capable identifying causes of milling operation faults, albeit with the exception of machine type errors, it omits to suggest an appropriate course of remedial action.

Research undertaken by Lee (1990) on data feedback in an integrated design to manufacture system forms the MDA part of the 'information support system for design and manufacture' project conducted in collaboration with Loughborough and Leeds Universities (Bell and de Pennington 1990, Corrigall et al 1992). The MDA activity provides the analysis of any deviated results determined by inspection and recommends appropriate actions should errors occur. This activity is supported by the product data model of the MOSES simultaneous engineering environment previously mentioned in chapter 3.5.2. This approach employs a feature-based concept where decision trees or networks associated with each feature operates by analysing measured data held within the measurement graph of the product model with the component nominals in order to classify them into one of three SPC type categories: (i) upper-fault, (ii) satisfactory and (iii) lower-fault.

The decision network is utilised to establish fault codes from an influence diagram. Each fault code is associated with a collection of information such as fault type, cause, action and probability located in a fault cluster of a fault library. The user is then presented with an ordered list comprising of the probable causes and suggested courses of action for each manufacturing error detected. The system developed is only
capable of analysing errors associated with individual features and is primarily directed to the prototyping stage of the product life-cycle.

Kramer and Nadanasundaram (1991) report another example of the application of a human interactive goal oriented backward chaining rule-based expert system. This system's problem domain is focused on the diagnosis of defects in milled components. Kramer and Nadanasundaram's system was constructed using the PC Plus expert system shell developed by Texas Instruments Corporation (Durkin 1994c). This prototype system can suggest corrective actions for common defects such as chatter, rough surface finish and dimensional inaccuracies. As mentioned previously in section 5.4.1, the milling error categories have been represented by four classes: workholding device, cutting tool, machine tool and general. This error structure is capable of diagnosing the causes of nineteen possible defects and is represented within a frame type knowledge base. The user initiates a consultation by the input of the observed defect into the expert systems. The execution procedure directs the user with a question and answer style dialogue from defect input through cause determination and remedy suggestion accompanied with an associated degree of certainty derived by the system from the user inputs. The system claims to be capable of identifying multiple causes for single and multiple defects and is directed at a user with little or no experience, however the strategy and method of diagnosis of the initial defect is not reported.

Another attempt at addressing the problem of interpretation of manufacturing errors from measured data obtained from the inspection of a component was established at Brunel University (Medland and Mullinuex 1993b, Medland et al 1994, Rentoul et al 1994). The approach conceived by Medland et al was validated with the aid of the previously mentioned RASOR constraint modeller (see chapter 4). With RASOR, tolerances extracted from the CAD system, are modelled as constraint rules which the component must not exceed if acceptable functionality and quality is to be achieved. RASOR attempts to solve these constraints to ascertain validity of the measured data (Mullinuex 1988, Medland and Mullinuex 1993b, Rentoul et al 1994). If all the constraints are evaluated as true then the problem is declared to be true and the component is within tolerance. On the other hand, if one or more constraint rules are untrue then any number of internal variables are allowed to vary to search for a scenario that is true. If this manipulation results in all the evaluated constraints to be true then the resultant values of the freed variable gives an indication of the magnitude of the manufacturing error. The main objective of the technique briefly outlined above is to
establish whether the inspection points correctly relate to the CAD model. If no match is found then a best-fit transform was determined which suggests an occurrence of a manufacturing error. In order to make a supposition regarding the possible sources of error a decomposition of the points was carried out (Rentoul et al 1994). According to Rentoul et al the decomposition of the manufacturing process could be represented by a simple hierarchical tree structure, which consisted of four levels:

i) part level; iii) tools level; and

ii) set-ups level; iv) features level.

The root of the tree hierarchy represents the completed manufacturing process. The next level represents the repositioning of the component or set-ups. The third level is associated with the tool used at each component set-up and the final level corresponds to the features produced by various tools in a certain set-up configuration. There are two possible methods in which the hierarchy can be employed to infer causes of manufacturing errors: a top-down; or a bottom-up approach (Rentoul et al 1994). If the manufacturing process is well established the top-down approach would be the most appropriate method as only one comparison is likely to find a match between the expected and the inspected data. If there is the prospect of analysing and obtaining inspection data concurrently then the bottom-up approach is more suitable as it facilitates data capture and analysis to be carried out simultaneously.

A rule-based expert system has been developed using the VP Expert system shell for the diagnosis of defects in plastic injection moulding (Luong et al 1997). The system basically consists of a dbase IV database, an inference engine, and a knowledge base, which is constructed using 47 production type rules. The system is capable of diagnosing one of a possible 10 production faults. The system relies totally upon the inputs supplied by the user to a set of predefined questions to direct the diagnosis.

Although the research reported above employs some facet of artificial intelligence to conduct the MDA activity it must be mentioned that there is a significant amount of research being conducted into the performance monitoring and final cut compensation schemes. This is achieved through the application of in-process part measurement technique and includes contributions from: Fan et al (1992), Mayer et al (1997), Yandayan and Burdekin (1997a, 1997b).
5.4.4 Critique – Machine Tool Error Diagnosis

The previous review of the diagnosis of manufacturing errors in relation to machine tools and manufacturing processes through the application of both condition monitoring and manufacturing data analysis illustrates the there is no standard or de facto methodology that can be applied in the troubleshooting problems in manufacturing. The solutions produced are directed to solve a basic number of manufacturing problems within an extremely narrow problem domain. Although condition monitoring of machine tools provide process control it is specifically geared toward the elimination of machine tool failure by the initiation of preventative maintenance. Manufacturing data analysis tries to achieve process control by inferring machine tooling errors form the inspection of manufactured parts. However, the application of manufacturing data analysis has been solely geared toward the prototyping phase of the product life-cycle thus negating the obvious product control benefit that can be gained as a consequence of the inspection process within a batch production environment. The author recognises that the combination of both methods of machine tool error diagnosis can provide mutual benefits that can be exploited within the manufacturing industry, although the systems would be complicated and difficult to maintain.
Chapter 6

THE PRODUCTION DATA ANALYSIS RESEARCH

6.1 Introduction

This chapter introduces the author's research work into a Production Data Analysis (PDA) framework. The research has been influenced by work of a research programme for the realisation of appropriate IT tools to improve the manufacturing performance of metalworking SMEs which forms the underlying foundation for the realisation of the PDA framework. The chapter is divided into two major sections: firstly it provides the context of the research by providing a summary of the EPSRC (GR/L27077) research programme. Secondly it introduces the novel PDA framework representing the author's major contribution. The functional and operational structure together with the information requirements of the individual elements of the PDA framework are described in chapters 7, 8 and 9 respectively.

6.2 EPSRC (GR/L27077) Project: IT Tools to Improve the Manufacturing Performance of Metalworking SMEs

The research reported in this thesis has been heavily influenced by the work carried out in the Department of Manufacturing Engineering at Loughborough University on the EPSRC (GR/L27077) project into the realisation of "IT tools to improve the manufacturing performance of metalworking SMEs" (Bell and Newman 1996). The project is targeted at a distinct group of SMEs which occupy the demanding and dynamic position at the end of the supply chain. These companies are required to be extremely agile in response to customer pressure and normally rely on meeting these demands by the use of considerable ingenuity and flexibility. This is usually achieved through informal communication protocols within a flat, non-hierarchical business structure and can either be categorised by a small sophisticated businesses using advanced technological equipment, or a small owner-managed business that depends on highly skilled personnel and conventional manufacturing equipment, or a combination of both.
The major assertions of the research project is two fold: (i) that the familiar information systems hierarchy and commercially available software tools that are employed within larger enterprises are inappropriate for adoption by the SME; and (ii) that the integration requirements of the SME are not satisfied by the scaling-down of these versions of deterministic structures, but require more appropriate support.

6.2.1 The Contemporary Small Manufacturing Enterprise

The traditional SME can be viewed as an enterprise that conducts its day-to-day business on a casual basis whilst exhibiting holonic characteristics with the minimum of information support (Koestler 1967, Suda 1989, 1990, Valckenaers et al 1994). The realisation of both autonomous and co-operative working practices, which underpin the fundamental concept of holonics, is achieved purely through human centred interactions. These human operators, in-turn, may have access to distributed stand-alone computer aided applications and associated hardware which are analogous to the islands of automation experienced within larger rigid CIM installations. The limitation of the current working practices in the exploitation of IT tools for SMEs has been summarised by Bell and Newman (1996) as:

i) the lack of a continuous information network being used between executive, business and manufacturing activities of the enterprise;
ii) the DNC network supports the bi-directional communication of the dominant node and the individual workstation nodes, but not between these individual workstation nodes;
iii) the SME is often supported by IT application tools that provide minimal support to human centred activities.

The focus of the research project is realised when the SME acquires appropriate IT tools. A holonic business model is proposed to represent the holonic characteristics of the IT supported SMEs manufacturing activities is depicted in figure 6.1. This model represents enterprise as an enhanced organisation holon which can further be categorised into three sub-holons namely: the executive holon that represents the ultimate decision-making process within the company, the business holon that covers administration activities such as order processing, finance, costing, process planning...
and scheduling etc., and the manufacturing holon involves the implementation and monitoring of the process plans produced by the business holon (Toh et al 1998).

Figure 6.1. The Conceptual Holonic Representation of an IT Supported SME (Bell and Newman 1996)

This organisation holon is seen to consist of a combination of human and workstation holons with each sharing the same interface. This is achieved through integrating both operators and workstations through an appropriate information network. This holonic information network (HIN) is designed using a SME reference model and provides the appropriate interfaces for operators and manufacturing workstations. Figure 6.2 represents the conceptual holonic business model, the HIN and the information resources captured by the SME reference model with regard to the manufacturing holon of the SME’s small jobbing shop environment.

Figure 6.2. The Implementation of the Holonic Business Model within a Metalworking SME (Bell and Newman 1996)
6.2.2 The SME Reference Model

As a consequence of the short comings identified in the utilisation of IT tools within the SME, Toh (1997, 1998) has established an enterprise model to provide the basis for realising the most pertinent IT tools to support improve the manufacturing performance of a typical metalworking SME. This model is capable of capturing the holonic and human centred characteristics of the SME and has been termed the SME reference model. The SME reference model is comprised from three fields of data, to wit: (i) organisation/behaviour; (ii) information and (iii) facility. The first field of data capture describes the organisation and behaviour, or holonic characteristics, of the enterprise. The second field of data involves the description of the information attributes related to the order, resources and process data. The third field of data represents the manufacturing facility description. These three fields of data correspond to three sub-models: (i) the organisation-behaviour sub-model; (ii) the order sub-model; and (iii) the manufacturing sub-model.

From the aforementioned three sub-model representation of the SME reference model, the order and the manufacturing sub-models are of major significance to this work reported in this thesis, a detailed discussion of which is reported in chapter 9.

6.3 The Production Data Analysis Framework

The author's research relates to the specific software of a node holon that constitutes a node on the holonic information network of the SME. The proposed Production Data Analysis (PDA) framework is specifically aimed to close the quality information feedback loop and support the multi-disciplinary, autonomous and co-operative working practices experienced within a contemporary holonically enhanced SME that cannot be achieved by the rigid scaled down versions of software applications employed within larger companies.

Figure 6.3 illustrates the interactions of the PDA framework with the information resources held within the SME reference model. In order to achieve the objectives of this research the PDA framework encompasses five vital issues that are considered essential to occupy the information feedback loop void that currently exists within contemporary manufacturing systems the distributed configuration of which is depicted in figure 6.4:
The five main functional elements of the PDA framework and the associated information interactions to produce effective feedback of both manufacturing performance and product quality are outlined in more detail below:

6.3.1 Machine and Inspection Planning

All of the research contributions into inspection planning to date have concentrated on the isolated generation of inspection plans and the automatic production of programs for execution on CMMs (Corrigall 1990, Menq et al 1989, Yau and Menq 1992, Rentoul et al 1994). Some contributions have centred on the variant approach, which involves the modification of an existing process planning and NC code generation facilities (Brown 1990, Tang and Davies 1990). Little or no consideration has been taken into account of the machining factors involved in producing the component in the first instance. These approaches can be extended to take into consideration the machining factors and parameters not only at the process planning phase, but also the inspection planning stage.

Although these tasks can be considered similar activities, they are treated as separate activities within the order sub-model. Both the operation and inspection plans are usually generated through mathematical simulation of cutter and probe paths.

Figure 6.3. PDA Information Resource Interactions
Figure 6.4. The PDA Framework’s Functionality within a Contemporary SME

The author’s proposed PDA framework is concerned with the simultaneous generation of operation and inspection plans from one unified source of information. This information should either be created interactively through a graphic user interface of an enhanced CAM system (Bagshaw and Newman 1998) or be extracted from data held within the order sub-model. Both the operation and inspection plans are concurrently verified through graphical simulation of cutter and probe paths.

6.3.2 Production Code Generation

As previously outlined, contemporary practice in production code generation usually involves the creation of an operation plan, operation simulation, the generation of set-up plans, and the post-processing of cutter location data into a machine dependent NC part program by the production planner. A similar procedure can be applied to the inspection planning task. However, this task is usually achieved via another planner responsible for inspection, whom may or may not reside within the same department. Although these tasks can be considered similar they are treated as two completely disjoint activities.

This PDA framework alleviates the requirement for two separate planners within two departments using disjoint software modules by allowing both the machine and
inspection planning activities, and subsequent production code generation to be conducted simultaneously and in parallel by a single planner. The planning, simulation and the automatic production of CMM inspection programs are achieved transparently to the planner.

6.3.3 Comparative Tolerance Analysis
The functionality of this phase is to compare the measured data, obtained from the inspection of a manufactured component on a CMM, against the nominal reference geometry and the design tolerance specification of the component obtained either from a CAD model representation or the order sub-model. The results of such an analysis are documented on a component status report which contains a detailed résumé of the analysis giving any out-of-tolerance deviations and nominal data regarding the associated feature. The component is classified into one of three categories accept, reject and rework, thus ensuring quality at a product control level.

The framework must support the automatic and reliable comparative tolerance analysis of manufactured components whilst appearing totally transparent to the planner. This allows the user to concentrate on rapid decision making and problem solving of unpredictable manufacturing disturbances.

6.3.4 Manufacturing Data Analysis (MDA)
The MDA functionality is based upon the outcome of the comparative tolerance analysis phase. This activity analyses the inspection results and the component status category obtained from the tolerance analysis report, ascertains the machine ID from the production plan held in the order sub-model and interrogates the machine dependent fault library information residing in the manufacturing model. The outcome of such an analysis will be:

i) machine dependent production error;
ii) probable cause for the production error; and
iii) recommended corrective actions to be taken for the machine in question.

The ascertained machine dependent production error, cause and action taken will be logged onto the manufacturing model's historical log for that machine. This information is subsequently utilised at the planning stage to ascertain the most
appropriate machine to undertake the operation or to initiate unplanned maintenance of the machine tool. This functionality provides the capability to initiate procurement of additional materials for rejected components, together with the rescheduling and regeneration of NC and inspection programs for re-workable components.

Based on the fundamentals of MDA, the concept of reactive manufacturing control can be realised through the effective information integration and feedback of the PDA framework together with the other manufacturing holons in the holonic manufacturing environment.

iv) **Data Resource Model Integration**

The information held within the PDA order and manufacturing sub-models consists of information captured and generated from the machine and inspection planning, comparative tolerance analysis and manufacturing data analysis phases of the PDA facility. These enhancements are the product of analyses of the data held within the generic order sub-model and the manufacturing sub-model thus closing the manufacturing information loop that exists between the available information resources of the enterprise. Therefore the PDA order and manufacturing sub-models enhance the effectiveness of enterprise information resources by integrating the information held within the generic order and manufacturing sub-model in order to produce manufacturing performance information from product quality data. The PDA order sub-model whilst capable of capturing a single instance of the component’s nominal geometry is also capable of capturing multiple instances of the component’s actual geometry that correspond to every instance of component inspection. This capability allows the captured actual information to be readily available for subsequent analysis by statistical process control applications.

6.4 **The Operational Structure of the Production Data Analysis Framework**

The operational structure of the proposed PDA framework has been modelled using the Integrated Definition method, more commonly known as IDEF0 (Meta Software Corporation 1995) (see chapter 3.8). This formal method of modelling was selected because of its ease of use and interpretation whilst enforcing strict rules of representation of information and activities in order to establish various levels of detail through a page hierarchy. An overall page and its decomposition sub-page of the IDEF0 model for the PDA framework is depicted in figures 6.5 and 6.6 respectively.
Figure 6.5. Overall IDEF0 Representation of the PDA Framework

Figure 6.6. The Sub-level IDEF0 Representation of the Operational Structure of the PDA Framework
Four main activities have been identified to fulfil the basic requirements of the prototype PDA framework: (i) concurrent operation and inspection planning which involves: create part geometry, create operation plan, and create inspection procedure, (ii) produce production code, (iii) comparative tolerance analysis, and (iv) manufacturing data analysis. These activities are modelled using the IDEF0 methodology and are described in detail in chapters 7 and 8.
Chapter 7

PRODUCTION DATA GENERATION AND INSPECTION ANALYSIS

7.1 Introduction
This chapter describes the production code generation and analysis phase of the PDA framework which is termed the Production Engineering Productivity System Inspection Module (PEPSIM). PEPSIM is primarily concerned with machine and inspection planning, NC part programming, CMM inspection code generation and comparative tolerance analysis. The sections covered within this chapter include:

i) Production Engineering Productivity System Inspection Module (PEPSIM);
ii) Machine and inspection planning;
iii) Operational structure of the machine and inspection planner;
iv) Production code generation;
v) Operational structure of the production code generator;
vii) Comparative tolerance analysis;
vii) Operational structure of the comparative tolerance analyser.

This chapter outlines the conceptualisation of the PEPSIM phase of the PDA framework followed by the operational structure and implementation of the prototype PEPSIM system. The remaining phase of the PDA framework, i.e. the manufacturing data analysis phase termed the Production data Analysis Distributed Diagnostic Expert System (PADDES), forms the subject of chapter 8.

7.2 Production Engineering Productivity System Inspection Module (PEPSIM)
The inspection phase (PEPSIM) of the overall production data analysis framework is highlighted in figure 7.1 and addresses the first three of the major issues identified as vital to the realisation of closed loop manufacturing. This phase termed PEPSIM constitutes three of the elements of the PDA framework consisting of:
i) machine and inspection planning,
ii) production code generation and
iii) comparative tolerance analysis.

7.2.1 Manufacturing Feature-based Characterisation

PEPSIM adopts the design by feature approach previously outlined in chapter 3.3 to represent the manufacturing features of a prismatic type component. PEPSIM's 2½ D feature library is based upon the feature taxonomy provided by the PEPS milling expert CAM system (see appendix I). These manufacturing features are represented as depression type features that are classified by the type of feature, which are in turn sub-categorised by the method of manufacture. The basic depression feature taxonomy of the PEPSIM system is illustrated in figure 7.2. The feature taxonomy consists of billet, hole pocket and key slot type features, which are further decomposed into the following specific features:

i) **Billet features** comprise of rectangular stock material from which the component is to be machined;

ii) **Hole features** include: drilled, reamed, bored, counter bored and socket head type holes;

iii) **Pocket features** consist of: closed blind rectangular, open blind rectangular, and round blind pockets;

iv) **Key slot features** include key slot open, key slot closed and key slot plunged.
Each of the features within the PEPSIM's manufacturing feature taxonomy is associated with a number of attributes that underpin the execution of the manufacturing and inspection activities of the PDA framework. These attributes and their associated generation and interpretation are discussed in the following sections.

### 7.3 Machine and Inspection Planning

This section outlines the powerful capabilities of the PEPSIM system to support the novel functionality of simultaneous and transparent generation and simulation machine operation and inspection plans. The focus is directed towards the essential requirements necessary for a commercially available NC part programming CAM system (see appendix I) in order to provide the ability to produce verified inspection plans and operation plans concurrently from a single source of information. This novel functionality is aimed at the contemporary metalworking SME and as such must be conducted with minimal effort and inspection planning expertise on behalf of the
operation planner. This application of tolerance information, probing point data and inspection probing paths determination are outlined in the following sections.

7.3.1 PEPSIM - Tolerance Representation and Interpretation

In order to perform the inspection task on a component a tolerance specification that describes the design intent must be captured within the PDA framework. The tolerance representation of an automated inspection systems must conform to the uniform practices for stating and interpreting dimensioning, tolerancing and related requirements for use on engineering drawing and related documents (ASME Y14.5M 1994, BS308 Pt.1 1984, BS308 Pt.2 1985, BS308 Pt.3 1972). The scope of this standard practice encompasses both dimensional and geometric type tolerancing principles. Whilst dimensional tolerances cover the dimensioning of feature size, geometric tolerances control the geometric characteristics of a component and include form, profile, orientation, location and run-out type tolerances. A complete categorisation of geometric tolerances and their application and symbology is depicted in table 7.1.

<table>
<thead>
<tr>
<th>For individual features</th>
<th>Type of tolerance</th>
<th>Characteristic</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>Straightness</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Flatness</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Circularity/Roundness</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cylindricity</td>
<td></td>
<td></td>
</tr>
<tr>
<td>For individual or related features</td>
<td>Profile</td>
<td>Profile of a line</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Profile of a surface</td>
<td></td>
<td></td>
</tr>
<tr>
<td>For related features</td>
<td>Orientation</td>
<td>Angularity</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Perpendicularity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Parallelism</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td>Position</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Concentricity</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Symmetry</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Run-out</td>
<td>Circular run-out</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Total run-out</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 7.1 Geometric Tolerance Characteristics (ASME Y14.5M 1994, BS308 Pt.3 1972)
Although the feature attributes held within the order sub-model (see chapter 9.3) are capable of all the dimensional and geometric tolerances stipulated within ANSI Y14.5M the feature tolerances information has been restricted to the plus and minus representation. As the subject of tolerances and tolerancing methods is an immense area of research in its own right, this restriction has been applied to dimensional tolerances and a limited selection of geometric tolerances namely: flatness, circularity/roundness, cylindricity, angularity and position. The tolerances for an PEPSIM’s open blind rectangular pocket feature are depicted in figure 7.3. and includes: size, position, roundness of corner radius and feature angularity.

Figure 7.3 Implemented Tolerances for an Open Blind Rectangular Pocket Feature

Due to the time constraints of this extensive research contribution a restriction has also been applied to the representation of datums. The datum reference frame for a rectangular type billet feature within PEPSIM is illustrated in figure 7.4 and defines the co-ordinated system and billet origin from which the machining and inspection are based. The component’s co-ordinate system origin lies on the top face and in the left-hand corner of the rectangular billet.

All construction features within PEPSIM are inserted on the top face of the rectangular billet. The datum faces of the billet feature are depicted in figure 7.5 and are employed to specify feature boundaries where the inserted construction feature interacts with both the top face and any side face of the billet. This representation is only required for key slot open, key slot plunged and open blind rectangular type features.
A datum feature reference has also been introduced to allow the insertion of a construction feature onto the base of an existing feature, although in the case of PEPSIM this has been restricted to a single level of insertion.

7.3.2 **PEPSIM - Feature Probing Point Distribution**

Feature probing points must be distributed uniformly over the feature's surface. Manufacturing features can be decomposed into geometric primitives such as: lines, planes, circles, spheres, cylinders and cones. The British Standard BS7172 (BS7172 1989) describes guidelines for the assessment of position, size and departure of nominal
form for geometric features and is directed in particular to software writers within the 
CMM industry.

This standard identifies strategies for the determination of the desired 
distribution and minimum number of probing points required in order to assess the 
position, size and departure of normal form of the aforementioned primitive features 
and is outlined in table 7.2. Although it is recognised that the distribution of probing 
point should normally aim for an even coverage of the workpiece, it is stressed that it 
should not be so regular that it is impossible to determine systematic or cyclic 
deformations such as the lobing of a circle. To avoid this problem it is suggested that a 
prime number of probing points at evenly distributed intervals around the primitive 
feature should be adopted.

<table>
<thead>
<tr>
<th>Geometric element</th>
<th>Minimum number of probing points</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Mathematical</td>
<td>Recommended</td>
</tr>
<tr>
<td>Line</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>Plane</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>Circle</td>
<td>3</td>
<td>7</td>
</tr>
<tr>
<td>Sphere</td>
<td>4</td>
<td>9</td>
</tr>
<tr>
<td>Cylinder</td>
<td>5</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>Cone</td>
<td>6</td>
<td>12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>15</td>
</tr>
</tbody>
</table>

Table 7.2 British Standard BS7172’s Recommendations for the Number of Probing Points for primitive geometric Features (BS7172 1989)

All the construction features represented within PEPSIM can be decomposed 
into a combination of the three primitive geometric elements: planes, circles and 
cyinders. The number of probing points employed by PEPSIM to inspect these three 
primitive geometric features is based on the BS7172 minimum requirements and is 
presented in table 7.3. A major issue in the selecting the probing points for each feature 
is that they are distributed uniformly over the entire area of the primitive geometric 
elements. The amalgamation of the construction feature’s geometric elements and their 
associated probing points provide the probing point distribution for that feature. The
probing point distribution and the exploded geometric element decomposition for all the features supported by PEPSIM is depicted in figure 7.6.

<table>
<thead>
<tr>
<th>Geometric element</th>
<th>Probing Point No.</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plane</td>
<td>4</td>
<td>Two lines of two for base planes for flatness</td>
</tr>
<tr>
<td></td>
<td>6</td>
<td>Two lines of three for other planes for flatness</td>
</tr>
<tr>
<td>Circle</td>
<td>4</td>
<td>One circle of four points for circularity/roundness</td>
</tr>
<tr>
<td>Cylinder</td>
<td>8</td>
<td>Two circles of four points for cylindricity</td>
</tr>
</tbody>
</table>

Table 7.3 PEPSIM's Probing Point Numbers

![Figure 7.6](image_url)

Figure 7.6 Manufacturing Feature Construction and Probing Point Distribution

The feature based probing point information and geometric element decomposition illustrated in figure 7.6 comprise:

i) **Hole feature** - Drilled through hole, Drilled blind hole, Reamed hole, Bored hole.

Geometric element – Cylinder, No. of probing points 8

Total number of probing points 8
**Hole feature** - Drilled flat hole, Round blind pocket (Diameter < 10 mm).
Geometric element – Cylinder, No. of probing points 8
Geometric element – Base Plane, No. of probing points 1
Total number of probing points 9

**Hole feature** - Drilled flat hole, Round blind pocket (Diameter >= 10 mm).
Geometric element – Cylinder, No. of probing points 8
Geometric element – Base Plane, No. of probing points 4
Total number of probing points 12

**Hole feature** - Counter bored hole, Socket head screw hole.
Geometric element – Hole Cylinder, No. of probing points 8
Geometric element – Bore Cylinder, No. of probing points 8
Geometric element – Base Plane, No. of probing points 4
Total number of probing points 20

**Pocket feature** – Closed blind rectangular pocket.
Geometric element – 4 x Side Plane, No. of probing points 24
Geometric element – Corner radius, No. of probing points 4
Geometric element – Base Plane, No. of probing points 4
Total number of probing points 32

**Pocket feature** – Open blind rectangular pocket.
Geometric element – 3 x Side Plane, No. of probing points 18
Geometric element – Corner radius, No. of probing points 4
Geometric element – Base Plane, No. of probing points 4
Total number of probing points 26

**Key slot feature** – Key slot closed
Geometric element – 2 x Side Plane, No. of probing points 12
Geometric element – 2 x End radii, No. of probing points 8
Geometric element – Base Plane, No. of probing points 4
Total number of probing points 24
Key slot feature – Key slot open, Key slot plunged
Geometric element – 2 x Side Plane, No. of probing points 12
Geometric element – End radius, No. of probing points 4
Geometric element – Base Plane, No. of probing points 4
Total number of probing points 20

As construction features that represent the geometry of the component are all inserted on the top of the billet feature the author has recognised that all the features in PEPSIM can be accessed using a single vertical probe orientation. Therefore an assumption of PEPSIM is that all feature probing points can be accessed by a single vertical tactile probe configuration thus negating the requirement for multiple probe orientations.

A limitation to PEPSIM’s algorithmic approach to feature probing point generation is that the probing points generated are fixed and cannot, at present, be modified by the user.

7.3.3 PEPSIM – Feature Based Probing Path Generation

The process of probing path generation is the activity of joining the generated probing points via movement vectors that dictate the movements of the probe during feature measurement.

As all the features supported by PEPSIM are depression type features, the geometric elements of the feature are internal and as such the initial and safe entry point of the probe into the feature cavity is directed to the centre of the feature which also corresponds to the insertion point of the feature on the billet (Camtek 1994). PEPSIM generates the probing path for each feature by inspecting in sequence the primitive geometric elements that are employed in its construction. The complete generated probing points and the associated probing path for a open blind pocket type feature is shown in figure 7.7. Each probing point of the feature has an approach and a retract distance associated with it along which the probe travels at the specified probing velocity. The end points of these distances are connected together for each individual geometric element of the feature. The intermediate paths between the approach and retract end points form the rapid velocity path of the probe. The combination of both path type form the complete path for that geometric element of the feature. The remaining portion of the probing path is created by joining the individual geometric
element paths together in sequence with a rapid velocity path to create the complete path for the feature. The final stage is to exit the feature to a safe position for traversing onto the next feature. In the case of a blind type feature this is achieved by traversing rapidly vertically from the last point which is always located on the feature base plane. However, with through hole type features, exiting of the cavity is achieved through the centre of the feature.

Collision avoidance when traversing between features is assured by employing a simple safety plane principle. The approach involves traversing the probe vertically on exit of the preceding feature to a safe plane some distance above the billet before moving to above the next feature. Entrance into the following feature is achieved vertically through the centre of the feature. PEPSIM allows the completed probing path for the component to be verified via a graphic user interface.

The novelty of PEPSIM's approach to inspection planning is that the probing points and probing path are generated by parameterised macros and as such are created completely transparently to the user negating the requirement for previous expertise and experience. This simple but effective methodology does not require any special computational requirements unlike the approaches adopted by Corrigall (1990), Spyridi and Requicha (1990), Menq et al (1989), Yau and Menq (1992) and therefore is ideally suited for application within a contemporary metalworking SME.
7.4 Operational Structure of the Machine and Inspection Planner

The author's proposed novel PDA framework is concerned with the simultaneous generation of operation and inspection plans from one unified source of information. This information can either be created interactively through a graphic user interface of an enhanced CAM system (Bagshaw and Newman 1998) or be extracted from data held within the order sub-model. Both the operation and inspection plans can be concurrently verified through graphical simulation of the cutter and probe paths.

The complete operational structure of PEPSIM has been modelled by the IDEF0 graphical activity modelling. The functionality of the concurrent machine and inspection planner element of PEPSIM is shown in figure 7.8 and primarily consists of the IDEF0 representation of three high level activities namely: create part model geometry, create operation plan and create inspection procedure.

7.4.1 Create Part Model Geometry

This activity, depicted in figure 7.8a, initially involves the creation of the billet from which the component is to be machined. The billet feature can consist of the any of the billet feature types previously mentioned, however, for the purposes of the PEPSIM phase of the PDA framework only the rectangular type billet feature has been investigated. Construction features are interactively defined and inserted in the appropriate orientation and location on the billet. During construction feature definition and insertion, the planner is prompted to specify a tolerance specification in accordance with ANSI Y14.5M and to assign an inspection priority for that feature. The feature creation and insertion procedure is graphically expressed in figure 7.9.

Once the planner has inserted the feature onto the billet, probing point algorithms generate a three dimensional point pattern that can be utilised to inspect the feature. It is at this point that the nominal feature attribute, which forms part of the order sub-model of the SME reference model (see chapter 9), is automatically populated with the general, feature nominals, tolerance specification and the probing point data. This procedure is repeated for each construction feature that constitutes the complete component. The geometric viewpoint of the order sub-model of the SME reference model nominal then comprises a complete geometric specification of the component.
Figure 7.8 IDEF0 Representation of the Concurrent Machine and Inspection Planning Functionality of the PDA Framework
(a) Create Part Model Geometry, (b) Create Operation Plan, (c) Create Inspection Plan
7.4.2 Create Operation Plan

This activity is essentially unchanged from the original commercial PEPS milling expert CAM methodology (see appendix I). Each feature type within the PEPS feature taxonomy (Camtek 1994) possesses its own parameterised machining operation procedure associated with it. These operations, tools, cutting speeds and feeds are assigned to the particular feature. The data regarding the tools, cutting speeds and feeds are obtained through rule based analysis of the feature and resource information contained within the manufacturing sub-model of the SME reference model. Once this has been achieved for each construction feature that constitutes the component, operations are sequenced to either tool optimisation or set-up optimisation criteria to form the complete machining procedure. The operation procedure produced by the
PEPS milling expert CAM system for an open blind rectangular pocket is illustrated in figure 7.10. This resulting procedure can be verified through machining simulation on the graphic user interface and a tool set-up sheet for the particular machine tool produced as illustrated in figure 7.11.

![Tool 11 FACE MILL ROUGH T25M D38 * 90 Operation PK 000 ROUGH .5/S](image1)

![Tool 16 SLOT DRILL M7 D16 * 40 Operation PK 000 SEMIFINISH .1/S 0/BTM](image2)

![Tool 12 NEW SLOT DRILL M7 D8 * 23 FINISH Operation PK 000 FINISH AROUND](image3)

Figure 7.10 Operation Sequence and Tooling Data for an Open Blind Rectangular Type Pocket

7.4.3 *Create Inspection Procedure*

This activity is initiated only if the inspection mode has been selected at the beginning of the component geometry construction stage. The first stage of this activity is to determine the sequence in which to inspect the construction features of the component. Although it is feasible to sequence the inspection to some criteria similar to the operation planning procedure, the author considered it acceptable for this research to restrict sequencing to the order in which the construction features were inserted.

With the inspection sequence determined, the relevant feature priority, nominal details and probing point co-ordinates are extracted from the nominal feature object of the order sub-model. This information is utilised by feature based simulation routines to ascertain a probing path, that includes approach and retract vectors, for each feature. Simulation of the complete probing path is achieved by the joining of these individual feature paths to form a continuous inspection path for the whole component. Collision avoidance of the probe tip and component is assured by ensuring both approach and
retract distances touch a safety plane away from the component surface of interest, removing the computational complexities of accessibility analysis reported by Spyridi and Requicha (1990) and Corrigall (1990). Simulation of the probing path can be graphically verified using the graphic user interface of the PEPS milling expert system, as shown in figure 7.11. The author has identified two classes of inspection based on: all feature inspection, and the identification of critical features of the component thus negating the need to undergo complete component inspection consisting of a large number of features.

![Figure 7.11 PEPSIM Machining and Inspection Simulation Capability](image)

(a) Cutting Tool Path Simulation, (b) Inspection Probing Path Simulation

### 7.5 Production Code Generation

As previously outlined, contemporary practice in production code generation usually involves the creation of an operation plan, operation simulation, the generation of set-up plans, and the post-processing of cutter location data into a machine dependent NC part program by the production planner (Camtek 1994) (see appendix I). A similar
procedure can be applied to the inspection planning task. However, this task is usually achieved via another planner responsible for inspection, whom may or may not reside within the same department. Although these tasks can be considered similar they are treated as two completely disjoint activities. This can be further emphasised by the modular structure of many commercially available CAM software suites. An exemplar is the EdgeCAM (Pathtrace Engineering Systems Ltd 1995) suite of modules. These modules can be procured separately or as a complete suite and as such little or no integration exists between the individual modules. In other words, one activity must be completed and the associated module terminated before the next activity can be executed.

This PDA framework alleviates the requirement for two separate planners within two departments using disjoint software modules by allowing both the machine and inspection planning activities, and subsequent production code generation to be conducted simultaneously and in parallel by a single planner. The planning, simulation and the automatic production of machine dependent and independent CMM inspection programs are achieved transparently to the planner.

The structure of the resultant inspection program produced by the inspection program generation routines is the same no matter what format is chosen. The structure of the inspection program contains the following sections:

i) \textit{CMM initialisation} involves the initial set-up of the CMM and includes: the inspection results file name, measurement units of mm or inches and measurement mode of automatic or manual.

ii) \textit{Probe tip calibration} involves the selection of the initial probing tip orientation which in the case of PEPSIM consists of a single vertical probing tip orientation consisting of a 20mm long shaft and a 2mm diameter ball end. Calibration of probe orientation is achieved under manual control by a single line of code for DMIS or a separate calibration file for HTBasic.

iii) \textit{Component alignment} is achieved manually in PEPSIM for each component using the six point, also known as the 3-2-1 alignment method for establishing the rectangular billet’s co-ordinate system. This involves manually taking three probing points of the top surface of the billet which identified the Z plane and restricts the Z plane’s origin and the rotation about both the X and Y axes. Two points are taken to create a line on the front side face that represents the X axis
which constrains the position of the Y axes and the rotation about the Z axes. A single point on the left end face that represents the Y axis which constrains the final degree of freedom, the position of the X origin. This procedure employed by PEPSIM does however assume that the surfaces of the rectangular billet are mutually perpendicular.

iv) Measure billet faces involves the measuring, under automatic control, of the billet's face planes in order to establish the billet's actual length and width and each plane's flatness. This is achieved by measuring four points on each of the top, front side, left end, rear side and right end surfaces of the billet.

v) Measure component features involves the automated measuring of each of the individual features in turn to the inspection plan, and uses the probing point distribution defined in section 7.3 and 7.4, which continues until all of the features in the inspection plan have undergone inspection.

vi) Output actual feature details is conducted in the case of the DMIS CMM program at the completion of the measurement of the manufacturing feature or at the end of the measurement of a primitive geometric element of the manufacturing feature for HTBasic programs. The output of the measurement for both program configurations is directed at providing both a hard copy of the results and a ASCII type text file representation.

The IDEF0 decomposition of the production code generator element of PEPSIM is shown in figure 7.12, the operation structure of which is described in the following section.

7.6 The Operational Structure of the Production Code Generator
This activity is responsible for the conversion of both the machine operation plan and inspection procedure into machine executable code. Machine NC part program generation employs the post processing facility which forms a powerful component of the PEPS milling expert CAM system and remains unchanged. The cutter location data is transformed to an executable machine dependent NC part program by a user generated post-processing program.

The generation of the CMM inspection code, however, is produced by a slightly different method. As with the simulation of the inspection probing path there exists a series of feature based inspection code generation routines that are responsible for the
Figure 7.12 IDEF0 Representation of the Produce Production Code Functionality of the PDA Framework
(a) Produce Production Code, (b) Produce CMM Inspection Program
generation of Dimensional Measuring Interface Specification (DMIS) or any other CMM dependent code, such as HTBasic, to inspect the feature and output the results as depicted in figure 7.13.

**Figure 7.13 DMIS Inspection Program and Plan Generation of PEPSIM**
All of the inspection programs produced by PEPSIM are capable of being executed on the CMM under servo control. Despite the complexity of the methodology employed in the creation of the production code, the execution of the above activities is completely transparent to the planner. As the data required for the execution of the program generation routines is extracted from the nominal feature attributes of the order sub-model (see chapter 9), all the planner requires to do is to generate the CAD model using construction features with the remainder of the production code generation procedure being achieved through menu selection.

7.7 Comparative Tolerance Analysis

The functionality of this phase is to compare the feature based measured data, obtained from the inspection of a manufactured component on a CMM, against the nominal reference geometry and the design tolerance specification of the component obtained from the order sub-model (see chapter 9). The results of such an analysis are documented on a component status report, which contains a detailed résumé of the analysis giving any tolerances, deviations and nominal data associated with the feature. The construction features and the overall component status are classified into one of three quality status categories regarding accept, reject and rework, thus ensuring quality at a product control level. An object oriented comparative tolerance analysis fact file is also produced through the execution of this phase of PEPSIM which provides the input to the manufacturing data analysis stage (see chapter 8), termed PADDES, of the PDA framework.

Although the novel production code generator of PEPSIM is capable of producing code in the standard DMIS format and the Ferranti CMM’s native HTBasic code format, the comparative tolerance analysis phase of the PDA framework has concentrated on the analysis of the inspection results produced from the execution of a HTBasic CMM program. This is due to the problem associated with the Ferranti CMM possessing an incomplete DMIS interpretor as expressed in section 3.5.4. The basic ASCII output information file produced from the execution of an automatically generated HTBasic is depicted in figure 7.14 and demonstrates that the output is presented in the form of the feature being decomposed into the actual basic geometric elements which form the feature i.e. planes, circles and cylinders (see table 7.3).

The comparative tolerance analysis of the billet in PEPSIM at present is purely directed to determining the actual length, width, size deviations, plane surface flatness...
and form deviation from nominal size and form of the rectangular billet. A typical documentation entry regarding the comparative tolerance analysis of a billet type feature is also depicted in figure 7.14.

![CMM Result File]

| REF_PLNA | X: 50.0124206956 | Y: 24.9831177512 | Z: -0.045125678639 | Flat: 0.00357678679 |
| REF_PLNB | X: 50.00816312881 | Y: .102363860682 | Z: -12.47063482052 | Flat: 0.0038276775 |
| REF_PLNC | X: .01869408666 | Y: 24.96909040792 | Z: -12.46034440782 | Flat: 0.0059687546 |
| REF_PLND | X: 50.0152773979 | Y: 50.0112385686 | Z: -12.4667559041 | Flat: 0.00264355667 |
| REF_PLNE | X: 10.024581152 | Y: 24.9272072204 | Z: -12.47820858019 | Flat: 0.00183773090 |

**Figure 7.14 Comparative Analysis of a Rectangular Billet Type Feature**

A more comprehensive approach to the comparative tolerance analysis has been adopted by PEPSIM for the analysis of the construction features. This approach is constant for all types of features supported by PEPSIM and consists of calculating the deviation from the nominal of each actual feature attribute, the feature attributes tolerance status i.e in-tolerance or out-of-tolerance and the feature’s attributes quality status i.e. accept, reject, rework.

The HTBasic CMM results consists of a sequential list of the measured data in the form of fitted primitive geometric elements shown in table 7.2 and are constructed by the fitting algorithms employed within the CMM’s software. In certain circumstances the actual feature dimensions attributes can be directly interpreted from the CMM results. This is the case for a basic through hole type feature as the hole can be represented by a single cylinder (see figure 7.6a) that corresponds to a single primitive geometric element within the CMM result file which provides the X, Y, and Z co-ordinates of the fitted cylinder that corresponds to the actual position of the hole, the diameter of the measured cylinder and the cylindricity of that cylinder. However
construction features that consist of a number of primitive geometric elements require some geometric reasoning. This reasoning entails the invisible creation of the actual construction feature on PEPSIM’s graphic user interface from the measured primitive geometric elements stated within the CMM’s result file. Once constructed standard PEPS CAD interrogation functions are employed to extract the actual construction feature’s attributes. The construction of the actual feature, with invisibility turned off, is shown in figure 7.15. The example shown in figure 7.15 is for illustration purposes and indicates that the actual open blind rectangular pocket feature is 5 mm smaller around the periphery of the pocket profile that its nominal counterpart.

Figure 7.15 Geometric and Comparative Tolerance Analysis of an Open Blind Pocket
The tolerance analysis is conducted by a simple comparison of the actual feature attribute to the limits imposed on the nominal feature attribute. The pseudo code representing the comparison is shown below illustrating how the tolerance status and quality status of a feature attribute are determined.

If $\text{attribute tolerance} > 0$ then

If $\text{act. feature attribute} < (\text{nom. feature attribute} - \text{attribute tolerance})$ then

Tolerance status = "OUTOL"
Quality status = "REWORK"

Else

If $\text{act. feature attribute} > (\text{nom. feature attribute} + \text{attribute tolerance})$ then

Tolerance status = "OUTOL"
Quality status = "REJECT"

Else

Tolerance status = "INTOL"
Quality status = "ACCEPT"

End if

End if

End if

The quality status feature attribute is calculated on the basis that all features within the scope of PEPSIM are of the depression type and as such if the actual feature attribute is outside the tolerance limit and smaller in comparison than the nominal attribute then the quality status of the feature attribute is designated as reworkable. Similarly if the actual feature attribute is outside the tolerance limit and larger in comparison than the nominal attribute then the quality status of the feature attribute is designated as rejectable.

Although the tolerance data representation captured within the order sub-model is capable of supporting all tolerance types stipulated within the ANSI Y14.5M dimensioning and tolerancing standard PEPSIM analyses all the feature attributes of a particular feature to verify dimensional tolerances, circularity/roundness tolerances, cylindricity tolerances, positional/location tolerances and angularity tolerances of angular sensitive features. A feature by feature breakdown of the tolerances supported by the comparative tolerance analysis phase of PEPSIM is outlined in table 7.4.
<table>
<thead>
<tr>
<th>Construction feature type</th>
<th>Type of tolerance ANSI Y14.5M</th>
<th>Feature attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic hole type feature:</td>
<td>Dimensional</td>
<td>Hole diameter</td>
</tr>
<tr>
<td>Drilled hole</td>
<td></td>
<td>Hole length</td>
</tr>
<tr>
<td>Bored hole</td>
<td></td>
<td>Hole length</td>
</tr>
<tr>
<td>Reamed Hole</td>
<td></td>
<td>Flatness</td>
</tr>
<tr>
<td>Round Blind Pocket</td>
<td></td>
<td>Hole position</td>
</tr>
<tr>
<td></td>
<td>Form</td>
<td>Hole cylinder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flatness (flat hole and round pocket only)</td>
</tr>
<tr>
<td>Compound hole type feature:</td>
<td>Dimensional</td>
<td>Hole diameter</td>
</tr>
<tr>
<td>Counter bored hole</td>
<td></td>
<td>Counter bore/sock</td>
</tr>
<tr>
<td>Socket head screw hole</td>
<td></td>
<td>Head diameter</td>
</tr>
<tr>
<td></td>
<td>Form</td>
<td>Hole cylinder</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flatness (counter bore/socket head base)</td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>Counter bore/sock</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Head hole position</td>
</tr>
<tr>
<td>Key slot type feature:</td>
<td>Dimensional</td>
<td>Key slot length</td>
</tr>
<tr>
<td>Key slot closed</td>
<td></td>
<td>Key slot width</td>
</tr>
<tr>
<td>Key slot open</td>
<td></td>
<td>Key slot depth</td>
</tr>
<tr>
<td>Key slot plunged</td>
<td></td>
<td>End circle 1 diam</td>
</tr>
<tr>
<td></td>
<td>Form</td>
<td>Flatness side planes A &amp; B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>End circle 1 circularity/roundness</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Flatness base plane</td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>Key slot position</td>
</tr>
<tr>
<td></td>
<td>Orientation</td>
<td>Key slot angularity (with ref. to billet x axis)</td>
</tr>
<tr>
<td>Pocket type feature</td>
<td>Dimensional</td>
<td>Pocket length</td>
</tr>
<tr>
<td>Closed blind rectangular</td>
<td></td>
<td>Pocket width</td>
</tr>
<tr>
<td>pocket</td>
<td></td>
<td>Pocket depth</td>
</tr>
<tr>
<td>Open blind rectangular</td>
<td></td>
<td>Corner circle diam</td>
</tr>
<tr>
<td>pocket</td>
<td>Form</td>
<td>Flatness side planes A &amp; B</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corner circle circularity/roundness</td>
</tr>
<tr>
<td></td>
<td>Location</td>
<td>Pocket position</td>
</tr>
<tr>
<td></td>
<td>Orientation</td>
<td>Pocket slot angularity (with ref. to billet x axis)</td>
</tr>
</tbody>
</table>

Table 7.4 Scope of the Feature-based Comparative Tolerance Analysis of PEPSIM
PEPSIM provides in the form of documentation a comparative tolerance analysis report document and a comparative tolerance analysis fact file. The comparative tolerance results comprises an ASCII text file which is constructed by appending the file with construction feature nominals, actuals, tolerances, deviations, tolerance and quality status after the analysis of each construction feature. The order sub-model is also populated simultaneously and transparently with the generated feature actuals. A fragment of a complete comparative analysis report document generated by PEPSIM is depicted in figure 7.16.

**Comparative Tolerance Analysis Report Document**

<table>
<thead>
<tr>
<th>Feature Details:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type</strong>: RECT</td>
</tr>
<tr>
<td><strong>Noms Acts Dev</strong></td>
</tr>
<tr>
<td><strong>Length (X)</strong>: 100</td>
</tr>
<tr>
<td><strong>Width (Y)</strong>: 50</td>
</tr>
</tbody>
</table>

**Insertion Point (X):** 50  
**Insertion Point (Y):** 25  
**Insertion Point (Z):** 0  

**PK_000 Feature Details:**  
**Feature Type**: POBK  
**Noms Acts Tols Dev** (vcr)

**Pocket Length:**  
**Lgh. Tol. Stat**: OUTOL  
**Lgh. Quality Stat**: REWORK

**Pocket Width:**  
**Wth. Tol. Stat**: OUTOL  
**Wth. Quality Stat**: REWORK

**Pocket Depth:**  
**Dhl. Tol. Stat**: OUTOL  
**Dhl. Quality Stat**: REWORK

**Corner Circle 1 Dia**: 10 | 9.9779 | 0.2 | -0.02208

**Dia. Tol. Stat**: INTOL  
**Dia. Quality Stat**: ACCEPT

**Location:**  
**Insert Point (X)**: 50 | 50.05095 | 0.15 | 0.05095

**X Pos Tol. Status**: INTOL  
**X Pos Quad. Status**: ACCEPT

**Insert Point (Y)**: 0 | 0.10296 | 0.15 | 0.10296

**Y Pos Tol. Status**: INTOL  
**Y Pos Quad. Status**: ACCEPT

**Insert Point (Z)**: 0 | -0.00451 | 0.15 | -0.00451

**Z Pos Tol. Status**: INTOL  
**Z Pos Quad. Status**: ACCEPT

**Orientation:**  
**Ref. About Z Axis**: 0 | 0.17276 | 0.002 | -0.17276

**Ang. Tol. Stat**: OUTOL  
**Ang. Quality Stat**: REJECT

**Form:**  
**Plane A Flatness**: 0.1 | 0.0146355

**Figure 7.16 Documentation Produced through the Execution of Comparative Tolerance Analysis Phase of PEPSIM**
The second form of documentation illustrated in figure 7.16 is an object oriented comparative tolerance analysis fact file. This fact file forms the input information for the final stage of the PDA framework, i.e. the novel manufacturing data analysis expert system known as Production data Analysis Distributed Diagnostic Expert System (PADDES) (see chapter 8) and is collated from the order sub-model upon completion of the comparative tolerance analysis. It describes the state of the component and is constructed to represent each feature as objects. The class diagram showing the class taxonomy expressed in Booch’s (1994) graphical representation is documented in figure 7.17 and reflects the feature taxonomy supported by PEPSIM (see section 7.2.1).

![Figure 7.17 Booch Class Diagram Expressing the Structure of the Comparative Tolerance Analysis Fact File](image)

Each feature object in the fact file contains all the feature nominals, actuals, deviations and status generated and reported within the comparative tolerance analysis report document with the addition of all the data regarding the operations employed to create the feature. This operation data is obtained from the nominal feature attribute portion of the order sub-model which is populated during the machine and inspection phase of PEPSIM (see chapter 9.3). The operation information is specifically directed to capture milling type operations such as drilling, boring and milling, each of which can be described in terms of operation tool data, operation parameters and an operation...
comment. For a detailed outline of the data attributes captured by the order sub-model the author directs the reader to chapter 9.3.

7.8 Operational Structure of the Comparative Tolerance Analyser
The operational structure of the comparative tolerance analysis of PEPSIM is documented in the IDEF0 representation of figure 7.18. The activity of comparative tolerance analysis can be decomposed into three major activities namely: comparative tolerance analysis of the billet feature which is conducted only once during a consultation, comparative tolerance analysis of each feature that constitutes the component and finally the compilation of both the documentation formats.

7.8.1 Comparative Tolerance Analysis of Billet Feature
The first major activity to be conducted by PEPSIM for component comparative tolerance analysis is to establish the deviations between the nominal dimensions and actual dimensions of the rectangular billet feature to which all construction features are referred to is functionally decomposed in figure 7.18b. This is achieved by initially creating a unique record within the order sub-model to capture the generated construction feature actual attributes. The billet nominals are then extracted from the construction feature nominal attribute portion of the order sub-model. For the purposes of this novel research only flatness tolerances have been applied to the plane surfaces of the rectangular billet feature. The next stage of the activity involves the extraction of the billet actual attributes from the CMM's result file, which consist of plane centre points and flatness values as illustrated in figure 7.14. The length and width of the actual billet are subsequently calculated and compared with their associated nominal values to ascertain discrepancies. These nominal and calculated actual billet attribute values are compiled and form the initial entry within the comparative tolerance analysis report document depicted in figure 7.14.

7.8.2 Comparative Tolerance Analysis of Construction Features
This activity of PEPSIM's comparative tolerance analysis phase is similar in structure to that employed for analysing the billet feature and is employed only once for each individual construction feature. However this activity is iterative in the sense that it is
Figure 7.18 IDEF0 Representation of the Comparative Tolerance Analysis
Functionality of the PDA Framework
(a) Comparative Tolerance Analysis, (b) Comparative Analysis of Billet,
Comparative Analysis of Feature, (d) Output Results
executed for each construction feature that appears in the inspection plan (see section 7.4.3).

The construction feature nominal attributes and tolerance specification conforming to that specified in the feature tolerance table 7.4 is extracted from the nominal portion of the order sub-model. The associated measured geometric elements for the construction feature are read from the CMM result file and if required the actual construction feature geometry is invisibly constructed and interrogated to ascertain the feature's attributes (see section 7.7).

These calculated actual feature attributes are subsequently compared with the feature nominal in order to calculate the deviations, tolerance conditions and status. A counter is initiated upon consultation and updated with every feature attribute analysis to monitor the overall quality status of the inspected component. These results are then compiled for the final stage of PEPSIM's comparative tolerance analysis phase, which is the output of the results.

7.8.3 Output of Comparative Tolerance Analysis Results

This is the final stage of PEPSIM's comparative tolerance analysis phase and is primarily concerned with both the administration of the resulting documentation and the population of the individual actual feature record of the order sub-model.

As mentioned previously the comparative tolerance analysis report document is constructed by appending a file with the construction feature attribute details during the analysis process. The procedure is duplicated for the population of the actual feature portion of the order sub-model. The construction of the comparative tolerance analysis fact file from the interrogation of the order sub-model, however, is completely constructed upon completion of the comparative tolerance analysis. The complete activity of comparative tolerance analysis is achieved without any intervention on behalf of the planner as all that the planner is required to specify is the type of CMM result file format at the onset of the consultation as in figure 7.19. This negates the requirement for additional inspection expertise and training and is directed toward the dynamic, multi-disciplinary working practices of the contemporary metalworking SME.

A single instance of the nominal attribute data regarding the construction feature is captured within the order sub-model during the creation of the component and multiple instances of actual attribute data or as measured and calculated information, that corresponds to each inspected component, is captured by the order sub-model at the
### Comparative Tolerance Analysis Report Document

**Billet Details:**

- **Type:** RECT

<table>
<thead>
<tr>
<th>Noms</th>
<th>Acts</th>
<th>Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length (X): 100</td>
<td>0.00589</td>
<td>0.00588706533</td>
</tr>
<tr>
<td>Width (Y): 50</td>
<td>0.009422805318</td>
<td>0.0094228053180032</td>
</tr>
<tr>
<td>Insertion Point (X): 50</td>
<td>0.00588706533</td>
<td></td>
</tr>
<tr>
<td>Insertion Point (Y): 25</td>
<td>0.0094228053180032</td>
<td></td>
</tr>
<tr>
<td>Insertion Point (Z): 0</td>
<td>0.0094228053180032</td>
<td></td>
</tr>
</tbody>
</table>

**PK_000 Feature Details:**

- **Feature Type:** POBR

<table>
<thead>
<tr>
<th>Noms</th>
<th>Acts</th>
<th>Tols</th>
<th>Dev</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pocket Length: 50</td>
<td>0.01869</td>
<td>0.1</td>
<td>-0.02225</td>
</tr>
<tr>
<td>Lgh. Tol. Stat.: OUTFOL</td>
<td>0.0059688</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lgh. Quality Stat.: REWORK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pocket Width: 35</td>
<td>0.0038277</td>
<td>0.1</td>
<td>-0.10919</td>
</tr>
<tr>
<td>Wth. Tol. Stat.: OUTFOL</td>
<td>0.0026436</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wth. Quality Stat.: REWORK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pocket Depth: 20</td>
<td>0.0045768</td>
<td>0.1</td>
<td>-0.00451</td>
</tr>
<tr>
<td>Dth. Tol. Stat.: OUTFOL</td>
<td>0.0018377</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dth. Quality Stat.: REWORK</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corner Rad.</td>
<td>0.01869</td>
<td>0.1</td>
<td>-0.02225</td>
</tr>
<tr>
<td>Dia. Tol. Stat.: OUTFOL</td>
<td>0.0059688</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dia. Quality Stat.: ACCEPT</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Location:**

- | Noms | Acts | Tols | Dev |
- Insertion Point (X): 50 | 0.01869 | 0.1 | -0.02225 |
- X Pos Tol. Status: OUTFOL |
- X Pos Qual. Status: ACCEPT |
- Insertion Point (Y): 0 | 0.10296 | 0.1 | 0.10296 |
- Y Pos Tol. Status: OUTFOL |
- Y Pos Qual. Status: ACCEPT |
- Insertion Point (Z): -0.00451 | 0.15 | 0.00451 |
- Z Pos Tol. Status: OUTFOL |
- Z Pos Qual. Status: ACCEPT |

**Orientation:**

<table>
<thead>
<tr>
<th>Feature Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature Type: POBR</td>
</tr>
<tr>
<td>Noms</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>Pocket Length: 50</td>
</tr>
<tr>
<td>Lgh. Tol. Stat.:</td>
</tr>
<tr>
<td>Lgh. Quality Stat.:</td>
</tr>
<tr>
<td>Pocket Width: 35</td>
</tr>
<tr>
<td>Wth. Tol. Stat.:</td>
</tr>
<tr>
<td>Wth. Quality Stat.:</td>
</tr>
<tr>
<td>Pocket Depth: 20</td>
</tr>
<tr>
<td>Dth. Tol. Stat.:</td>
</tr>
<tr>
<td>Dth. Quality Stat.:</td>
</tr>
<tr>
<td>Corner Rad.</td>
</tr>
<tr>
<td>Dia. Tol. Stat.:</td>
</tr>
<tr>
<td>Dia. Quality Stat.:</td>
</tr>
<tr>
<td>Insertion Point (X): 50</td>
</tr>
<tr>
<td>X Pos Tol. Status:</td>
</tr>
<tr>
<td>X Pos Qual. Status:</td>
</tr>
<tr>
<td>Insertion Point (Y): 0</td>
</tr>
<tr>
<td>Y Pos Tol. Status:</td>
</tr>
<tr>
<td>Y Pos Qual. Status:</td>
</tr>
<tr>
<td>Insertion Point (Z): -0.00451</td>
</tr>
<tr>
<td>Z Pos Tol. Status:</td>
</tr>
<tr>
<td>Z Pos Qual. Status:</td>
</tr>
</tbody>
</table>

**Pocket Details:**

- **Pocket Length:** 35 |
- **Pocket Width:** 20 |
- **Pocket Depth:** 20 |

**Orientation:**

- **Orientation:**

---

**Figure 7.19 PEPSIM's Comparative Tolerance Analysis Phase Initiation**
comparative tolerance analysis stage. Multiple instances of the comparative tolerance analysis fact file that also correspond to each inspected component are produced by 'outputting of the results' activity of the comparative tolerance analysis phase.

7.9 Summary of Production Code Generation and Inspection Analysis
To summarise, this section has introduced the concept and operational structure of the PEPSIM activities of the PDA framework. PEPSIM encompasses the activities of machine and inspection planning, production code generation and comparative tolerance analysis which is primarily aimed at the multi-disciplined operation activity conducted within a contemporary metalworking SME. The total production code and inspection analysis documentation is summarised in figure 7.20 and includes: tool set-up sheet and NC part program, which form the contributions to the PDA framework of the original PEPS milling expert CAM system, inspection plan. The CMM inspection program, comparative tolerance analysis report and comparative tolerance analysis fact files provide novel functionality to the CAM system, which in its entirety forms the PEPSIM phase of the PDA framework.

The final activity of the PDA framework is to conduct manufacturing data analysis in order to ascertain possible production causes from the comparative tolerance analysis activity of PEPSIM. This is achieved through the application of a rule-based expert system known as PADDES which forms the subject of the following chapter.
Set-Up Sheet

Tool Number 1 Block at Tool Change 00 ' Tool Type : SLOT DRILL M7 D28' 63 Tool Full Diameter: 28.0 Tool Length 63.0 NC Part Program

PK 006 ROUGH 5/6 Spindle Speed 660 rpm Feed Horizontal 72.8 mm Feed Vertical 18.2 mm

PK 001 ROUGH 5/6 Spindle Speed 2737 rpm Feed Horizontal 65.6 mm Feed Vertical 164.2 mm

---

NC Part Program

N05 SET PROG,TS5001,DEMO1)
N08 (**) JOB SIDE 1)
N10 (**) JOB SIDE 1)
N20 (TOOL 1 SLOT DRILL MT D28 '63)
N30 (OFFSET LENGTH 1 RADIUS 1)
N40 T1 M6
N50 D1 E10
N60 (PK 006 ROUGH 5/6)
N70 X36.4 M3
N80 G0 X60.0 Y80.0
N90 G0 X75.5 Y80.0
N100 G0 Y74.5 Z10.0
N110 G0 Y74.5 Z72.8
N120 X75.5 Y80.0 Z10.0
N130 X83.5 Y80.0
N140 X45.4 Y80.0
N150 X83.5 Y80.0
N160 Y74.5
N170 X45.4 Y80.0
N180 D1 M6
N190 D6 E16
N200 H(0.000 UDRILL)
N210 S2292 M3
N220 D6292292 M3
N230 X30.0 Z30.0 E298.0
N240 Z0.0 G0 Z100.0
N250 D810 (TOOL 1 SLOT DRILL)
N260 T20 M6
N270 X30.0 Y30.0 Z3.0
N280 G0 Z100.0
N290 X30.0 Y30.0 Z3.0
N300 X30.0 Y30.0 Z3.0
N310 M9
N320 (TOOL 1 SLOT DRILL)

---

Inspection Plan

Inspection Plan EXPLI 1.pJm

Date Thursday 12th February 1998

Summary Information:

Part Number: EXPLI

Inspection Plan Mode: All Feature Inspection

---

HTBasic CMM Inspection Program

5 PART: SUB Part
10 ! Acessat (HTBasic) Inspection Program
15 ! Program Name... EXPLI
20 !
25 ! COM /C2/X, Y, Z, R, A, D, D2, Tptr4Form, Pts(), [k-, (.)
30 ! COM /C5/Par(), Tp(), Pft())
35 ! COM /C8/Ln(), Cv(), Pln(), Cy!(), Sph()!
40 ! COM /C1 I/Op$, Opl$, Op2$, Op3$, Printer, Pnntr$, PtIg !
45 ! COM /C14/R 3d(), R 2d(), W()
50 ! COM /G res/Crace(), Invol(), Eccen(), Ctunul(), Adlaa): Ruck()
55 ! COM /Res/HdrS(), Sn!Prmlo$ "ON" ! For Geom Tol Prmtout
60 !
65 !
70 ! Start I:
75 Start I... .
80 !
85 Manual
90 Rot clr
95 Sel tip()!
100 H$[1]="NAME "-EXPLI "
105 H$[2]="NAME "-EXPLI"
110 !
115 !
120 !
125 !
130 !
135 !
140 Units["MM"
145 !
150 !
155 !
160 !
165 !
170 !
175 !
180 !
185 !
190 !
195 !
200 !
205 !
210 !
215 !
220 !
225 !
230 !
235 !

---

Comparative Tolerance Analysis Fact File

definitions BILLET OBJ BILLET (BILLET of BILLET) (component_id EXPLI 1)
(feature names BILLET) (feature_type RECT) (nom billet lgh 100)
(nom billet wth 100)
(nom billet dth 50)
(set act lgh 100.01068 )
(set act wth 100.01561 )
(set act dth 100.02208 )
(ref phlb xpos 0.0017291 )
(ref phlb ypos 0.0018254 )
(ref phlb zpos 0.000627 )
(ref phlb zpos 0.000411 )
(ref phlb xpos 0.0012065 )
(ref phlb ypos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )
(ref phlb zpos 0.00161412 )

---

Comparative Tolerance Analysis Report

Billett Details:

Billett Type: RECT
Billett Length (X): 100
Billett Width (Y): 100
Billett Depth (Z): 50

---

Figure 7.20 Production Code and Documentation provided by PEPSIM
Chapter 8

MANUFACTURING DATA ANALYSIS AND DIAGNOSIS

8.1 Introduction
This chapter describes the final manufacturing data analysis phase of the PDA framework termed the Production Data Analysis Distributed Diagnostic Expert System (PADDES). PADDES is responsible for the machine dependent diagnosis of manufacturing errors from the analysed inspection data generated through the PEPSIM phase of the PDA framework. The source of knowledge acquisition, knowledge representation and the inference strategy employed by the PADDES system are fully explained. The chapter concludes with a description of the operational structure of the novel PADDES system.

8.2 Production Data Analysis Distributed Diagnostic Expert System (PADDES)
The manufacturing data analysis functionality of PADDES is based upon the outcome of the comparative tolerance analysis phase, together with the analysis of the inspection results facts automatically generated by PEPSIM. PADDES utilises the machine dependent knowledge base rules, contained within the manufacturing sub-model of the SME reference model in a forward chaining process to infer production errors, and remedial actions from the inspection of 2½ D prismatic parts. The ascertained machine dependent production error, cause and action taken is logged onto the manufacturing model’s historical log for that machine. This information is subsequently utilised at the planning stage to ascertain the most appropriate machine to undertake the operation or to initiate unplanned maintenance of the machine tool and to plan reclamation work.

8.2.1 Expert System Development Tool
The PADDES phase of the PDA framework employs the C Language Integrated Production System (CLIPS) version 6.1 expert system tool, developed originally by NASA (Giarratano 1998, Giarratano and Riley 1998, CLIPS 1998), to conduct the manufacturing data analysis activity. This expert system shell (see chapter 5.3.2) provides a forward chaining inference strategy based on the Rete algorithm that is
CLIPS employs two programming paradigms: procedural programming and object-oriented programming the latter of which is termed the CLIPS Object-Oriented Language (COOL). CLIPS provides support for the development of modular programmes whilst providing tight integration between its object-oriented and rule-based programming capabilities. Since its development CLIPS has received widespread acceptance throughout government, industry and academia due to its portability, extensibility, capabilities and low cost. CLIPS powerful capabilities has helped to improve the ability to deliver expert system technology throughout both the public and private sectors for a wide range of applications within diverse computing environments.

**Figure 8.1 PDA Framework Functional Structure**

PADDES performs the diagnosis of manufacturing errors from the inspection results generated by the PEPSIM phase of the PDA framework by utilising all the procedural, rule-based and object-oriented capabilities provided by the CLIPS expert system shell. The following sub-sections of this chapter describe the facts and knowledge representation adopted by PADDES in its diagnosis.

### 8.2.2 Component Fact Representation for PADDES

The structure of PADDES follows the standard expert system configuration previously mentioned in chapter 5.3.2 and consists of the major elements as depicted in figure 8.2
(Durkin 1994a): a working memory, a knowledge base and an inference engine. PADDES employs COOL's object-oriented fact representation to portray the actual state or condition of the component and are generated from the order sub-model during PEPSIM's comparative tolerance analysis of the PDA framework. These facts comprise the input to PADDES and are temporarily stored within the short term or working memory of the expert system. As mentioned in the previous chapter, this comparative tolerance analysis fact file comprises feature class definitions, the taxonomy of which is shown in figure 7.17, and the instances of the classes that correspond to the inspected construction feature objects of the component.

The COOL object representation of an open blind pocket type construction feature which is an instance of the O_B_R_POCKET class is depicted in figure 8.2. The instance definition captures not only the nominal and actual feature attributes but also the information regarding the operations employed in the production of the feature, such as operation tool geometry, operation parameters, and operation comments. As the information contained within the comparative tolerance analysis fact file provides the complete and comprehensive fact information required by PADDES in order to conduct a diagnostic consultation. Since there is absolutely no requirement for interaction with the diagnostic consultant the forward chaining inference strategy (see chapter 5.3.3) provided by the CLIPS expert system shell's inference engine proves to be ideal for the diagnostic task.

8.2.3 Knowledge Acquisition for PADDES

Although there are numerous methods of acquiring the knowledge to emulate the human expert (see chapter 5.3.2.3), the method adopted in this research is that of extracting the required generic information from manufacturing handbooks and tooling catalogues. These include the Society of Manufacturing Engineers' publications: Tool and Manufacturing Engineers Handbook (Drozda and Wick 1983) and Troubleshooting Manufacturing Processes (Gillespie 1988). The manufacturing process information presented by these publications is comprehensive, however the specific information extracted for use by PADDES has been restricted to the generic manufacturing operations that can be conducted on a three axis type milling machine. These operations include: milling, drilling, boring, and reaming operations that correspond to the operation type that are supported by the powerful PEPSIM phase of the novel PDA framework.
Figure 8.2 PADDES Expert System Working Memory Representation
(Adapted from Durkin 1994a)
The author acknowledges that this generic operation information is applicable for any type of milling process and as such can be applied to diagnose manufacturing errors produced on any type of three axis milling machine. To increase the knowledge of PADDES to diagnose manufacturing errors for a particular milling machine, i.e. a 3-axis Wadkin vertical machining centre, specific machine information must be acquired from an additional expert source. This shallow knowledge is based upon local experience that can be acquired through the knowledge elicitation from the human expert through applying interviewing techniques to experienced shop floor operatives. Although the author realises that both forms of knowledge are invaluable when indulging in a diagnostic consultation, only the published type of knowledge has been implemented within PADDES. This is primarily due to the scope of the research and time constraints imposed on the author.

8.2.4 PADDES Manufacturing Error Categorisation

A comprehensive categorisation of the types of manufacturing errors regarding the milling process is expressed in figure 8.3. This categorisation is based upon the error classifications identified in chapter 5.4.1 and consists of: cutting tool errors, fixturing and work holding errors, machine tool errors, miscellaneous errors, and a completely new area of programming errors.

Figure 8.3 Feature-based Production Errors Applicable to the Milling Process

The milling production errors are further classified into categories that effect the geometric characteristics of a component. These categories include the type of error that can effect individual features and those that effect all the features of a component.
i) *Individual feature errors* are those that affect only one feature and include: cutting tool errors such as tool size error, tool run-out/misalignment error, tool wear and tool deflection. Programming errors such as feature size error, feature position/orientation error and interpolation error; and finally miscellaneous errors that relate to cutting conditions such as chatter and workpiece deflections.

ii) *Combined feature errors* comprise of the errors that propagate through the entire component and involve: machine tool errors such as axis out-of-calibration errors, servo lag/interpolation errors, stiffness errors, thermal distortion errors and random or stochastic type errors. Fixturing and workpiece deflection errors include: set-up errors between component and machine, fixture and machine and component and machine interfaces and insufficient chip control. Miscellaneous error arising from dimensional errors of the stock material.

The approach adopted by PADDES whilst conducting a consultation is directed towards the attainment of the individual feature errors of cutting tool type and programming type for each attribute of a feature and for all subsequent features that constitute the component part. The manufacturing data analysis of PADDES is only initiated if the feature attribute contained within the comparative tolerance analysis fact file under scrutiny possesses a defective quality status of either REJECT or REWORK. A second analysis phase of PADDES is concerned with the determination of possible combined feature errors of machine tool type, out-of-calibration of axes in particular, and component set-up error.

8.2.5 *Knowledge Representation of PADDES*

The knowledge extracted from the Tool and Manufacturing Engineers Handbook and the Troubleshooting Manufacturing Processes publications can be represented in the form of decision or logic trees. These tree diagrams provide a graphical portrayal of the logic embedded within the long-term memory, known as the knowledge base, of the PADDES expert system. Each feature geometric attribute of every feature supported by PEPSIM has a decision tree associated with it in PADDES. At present the implementation of logic to analyse manufacturing errors from feature form type geometric anomalies are not supported by PADDES. The complete decision tree representation for all features and their associated attributes are displayed in appendix III.
A decision tree for diagnosing manufacturing errors relating to the diameter of a drilled hole type feature is shown in figure 8.4. The decision tree illustrates how the logic captured by the knowledge base is applied to the component feature facts held within the working memory of PADDES.

![Decision Tree Diagram](image)

**Figure 8.4 Logic Decision Tree for Diagnosis of Diameter Errors of a Drilled Hole**

The logic embedded within the knowledge base first interrogates the fact file to ascertain whether or not the feature attribute's quality status is acceptable or not. If the attribute is acceptable no further action is taken for that attribute and PADDES then progresses to the next attribute in question. However, should the quality status of the feature attribute, in this case drilled hole diameter as in figure 8.4, be either REJECT or
REWORK then PADDES attempts to acquire the probable cause and subsequent remedy. PADDES then compares the tool diameter for the final drilling operation and compares it to the required nominal hole diameter. The result of the comparison directs the consultation through the appropriate branch of the decision tree. If the drill diameter is equal to the required diameter then it can be deduced that the correct tool was specified within the part program and that there must be a tool anomaly on the machine tool. The branch of the decision tree continues the diagnosis when the correct tool used in the program is established. PADDES tests the hole diameter and categorises the diameter into three ranges i.e. 0-6 mm, 6-19 mm and 19 and above. This is in accordance to the average accuracy ranges specified for twist drills from the Troubleshooting Manufacturing Processes handbook (Gillespie 1988). The complete listing of all feature and associated feature attributes investigated by PADDES is shown in table 8.1.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Feature Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hole Drilled/Reamed/Bored</td>
<td>Diameter</td>
</tr>
<tr>
<td>Counterbored Hole</td>
<td>Hole Diameter</td>
</tr>
<tr>
<td></td>
<td>Counterbore Diameter</td>
</tr>
<tr>
<td></td>
<td>Counterbore Length</td>
</tr>
<tr>
<td>Socket Head Screw Hole</td>
<td>Hole Diameter</td>
</tr>
<tr>
<td></td>
<td>Bore Diameter</td>
</tr>
<tr>
<td></td>
<td>Bore Length</td>
</tr>
<tr>
<td>Key Slot Closed</td>
<td>Combined Length &amp; Width</td>
</tr>
<tr>
<td>Key Slot Open</td>
<td>Depth</td>
</tr>
<tr>
<td>Key Slot Open</td>
<td>Orientation</td>
</tr>
<tr>
<td>Pocket Open Blind Rect.</td>
<td>Combined Length &amp; Width</td>
</tr>
<tr>
<td>Pocket Closed Blind Rect.</td>
<td>Depth</td>
</tr>
<tr>
<td></td>
<td>Orientation</td>
</tr>
<tr>
<td>Pocket Round Blind</td>
<td>Diameter</td>
</tr>
<tr>
<td></td>
<td>Depth</td>
</tr>
<tr>
<td>ALL Features</td>
<td>X, Y, Z Position</td>
</tr>
</tbody>
</table>

Table 8.1 PADDES Individual Feature Attribute Coverage
The results of such an analysis are then asserted into the working memory of PADDES for scrutiny by the second phase of the manufacturing data analysis. These cause and action scenarios constitute the leaves of the decision tree. It must be noted that, in the case of the decision logic for the diameter of a drilled hole, the scenario of a smaller diameter hole tool wear was not investigated as drills tend to drill larger diameters when a significant amount of tool wear is present. The approach is repeated for every attribute of every feature within the fact file. All possible connotations that can arise as a consequence of the tests undertaken at each branch of the tree are catered for thus assuring that any errors identified through the comparative tolerance analysis of PEPSIM will be associated with a probable cause and remedial actions.

Once all the feature attributes for all features have been investigated the second phase of PADDES is initiated. This phase is primarily concerned with the identification of component set-up and machine out-of-calibration errors that effect all features present on the component. This phase analyses the asserted probable causes of positional errors generated from the individual feature manufacturing data analysis. The combined feature decision trees and for the whole PADDES knowledge base is situated in appendix III.

The decision trees are represented within the PADDES knowledge base in the form of IF and THEN type diagnostic rules as illustrated in figure 8.5, a breakdown of which is provides in figure 8.6. One of the characteristics of expert systems is the rules captured in the knowledge base can be cited in any order as the execution of the rules is purely dependent upon the facts in the working memory satisfying the conditions or antecedents of the rules. In PADDES the precedence of rule execution is tightly controlled in a number of ways. The first method is the application of a salience value in the rule header, see figure 8.6, ranging from +1000 to -1000, to prioritise their importance. The former being of highest priority whilst the latter being of the lowest. In PADDES this method is used to control the execution of the rules in a feature oriented manner. The particular assigned saliences to feature rule are as follows:
Figure 8.5 Rule based Knowledge Base Representation of PADDES
Both length ACCEPT and width SMALLER

(defrule Pock1210
  (object (is-a O_B_RPOCHET))
  (feature_name ?name)
  (feature_type ?typ)
  (pockr-width-deviation ?wthdev)
  (pocket-length-condition ?nme ?typ)
  (rule_number 1210)
  (rule_salience -400)
  )

?ans <- (object (is-a O_B_RPOCHET))
  (feature_name ?name)
  (feature_type ?typ)
  (pockr-width-deviation ?wthdev)
  (pocket-length-condition ?nme ?typ)

?ans <- (object (is-a O_B_RPOCHET))
  (feature_name ?name)
  (feature_type ?typ)
  (pockr-width-deviation ?wthdev)

?ans <- (object (is-a O_B_RPOCHET))
  (feature_name ?name)
  (feature_type ?typ)
  (pockr-width-deviation ?wthdev)

?ans <- (object (is-a O_B_RPOCHET))
  (feature_name ?name)
  (feature_type ?typ)
  (pockr-width-deviation ?wthdev)

?ans <- (object (is-a O_B_RPOCHET))
  (feature_name ?name)
  (feature_type ?typ)
  (pockr-width-deviation ?wthdev)

printout t "Pocket Open Blind Rect. Width Too Small" printout "Pocket Width Deviation:" ?wthdev" mm"

printout t "1. Check interpolation path of tool for pocket operation." crlf
printout t "2. Regenerate NC Part Program." crlf

PADDES resolves rule-firing conflict of activated rules with identical saliences in one of two ways: The first method involves the firing of the most general rules before the more specific rules. In other words the rules that contain the least number of condition elements i.e. general rules, within the antecedent or IF component fire first, see figure 8.6. The rules fire in general to specific order until the final most specific rules fire. PADDES then fire the activated rules with the next highest salience priority. This method, however does not resolve the firing conflict of rules possessing identical saliences.

Figure 8.6 A Breakdown of an Open Blind Rectangular Pocket Diagnostic Rule

i) Billet +900 vii) Key Slot Closed 0
ii) Drilled Hole +500 viii) Key Slot Open -100
iii) Reamed Hole +400 ix) Key slot Plunged -200
iv) Bored Hole +300 x) Closed Blind Pocket -300
v) Counterbored Hole +200 xi) Open Blind Pocket -400
vi) Socket Head Hole +100 xii) Round Pocket -500

All features of the same type will be analysed simultaneously before analysis of the subsequent type feature is investigated. The combined feature analysis possesses a salience of -600 and under thus ensuring that all individual feature analyses are complete prior to execution.
saliences and number of condition elements. PADDES's final and precise method of controlling rule firing is achieved through fact manipulation, which involves the addition of a new fact to the fact list in the working memory. This is conducted by asserting a new fact in the consequent or THEN component of the rule as in second statement figure 8.6. This newly asserted fact can then be placed within the antecedent part of the rule that is required to fire next. This controlled rule is then executed always after the preceding rule. To avoid the accumulation of unnecessary facts within the working memory's fact list the controlling fact can be subsequently removed or retracted from list. This is achieved by the addition of the retract statement in the consequent of the controlled rule as in first statement figure 8.6.

All of these methods of conflict resolution are employed within PADDES to control the execution of the manufacturing data analysis whilst ensuring consistent rule firing order. At present the PADDES diagnostic knowledge base contains approximately 2000 diagnostic rules which cover the eleven 2½ D depression type features of PEPSIM.

8.2.6 PADDES Consultation Documentation

The documentation produced by a consultation with PADDES is provided in the form of a (.paddes) file and a machine tool historical log as shown in figure 8.7. The production data analysis distributed diagnostic expert system file provides a complete record of the consultation analysis from start to finish including both the individual feature error and combined feature error diagnosis of the machine dependent knowledge and is based on the component's comparative tolerance fact file. This assists the user, not only in the rapid identification of probable manufacturing error and the explanation of the appropriate courses of action, but also provides the user with a step by step explanation of how the reasoning of PADDES attained the conclusions. The machine dependent historical log provides an updateable file that is appended with any machine type errors that may have been identified by the combined feature error analysis of PADDES. This file consists of a breakdown of the machine tool specification obtained from the manufacturing sub-model. Any machine errors encountered during the consultation are appended to the end of the file. This file is intended to provide the planner with an instant record of the current and past errors and machine condition and status which can be applied to influence future routing decisions.
PADDES Analysis File

Feature [BILLETT] Feature Name BILLETT

** PK_000 Pocket Pos. in X OK POBR **

** PK_000 Pocket Pos. in Y OK POBR **

** PK_000 Pocket Pos. in X REJECT POBR **

** PK_000 Pocket Pos. in Y REJECT POBR **

PK_000 Pocket Open Blind Rool. Position in X not acceptable
Pocket Pos. in X Deviation = -0.10295999

For Pocket Open Blind Rool. PK_000 Pocket Out Of Position in X Axis by: 0.10295999 mm
Programming/Part Set-up Error

** ERROR CAUSE **

** ERROR CORRECTION **

1. Check pocket milling position in part program with specification and modify as required.
2. Clean and realign billet set-up for subsequent parts.

PADDES Machine Historical Log

Date Saturday 7th August 1999
Summary Information
Part Number: POBR2
Part File Name: POBR2 MIL
Part NC Tape File Name: POBR2 TAP
Part HTBasic File Name: POBR2 X
Tip Calibration File Name: POBR2 C

COMPONENT IDENTIFICATION No: POBR21

COMBINED FEATURE ERROR CAUSE

MACHINE ERROR
All Feature X Position Error Deviations are Equal
Feature X Positional Error Average = 0.43657 mm
MACHINE IS OUT OF CALIBRATION IN THE X AXIS!

ERROR CORRECTION

1. Check Component/Feature set-up interface in Z Axis
2. Check Component/Machine set-up interface in Z Axis
3. Check Machine Origin Position in X Axis
4. Conduct a Static Alignment Test of the Machine Tool in the X Axis Direction

Figure 8.7 PADDES Manufacturing Data Analysis File and Machine Historical Log Documentation

163
8.3 Operational Structure of Production Data Analysis Distributed Diagnostic Expert System

The IDEF0 decomposed structure of the PADDES expert system is illustrated in figure 8.8. The figure decomposes the manufacturing data analysis activity into: obtain component feature fact file, obtain machine dependent fault library, conduct manufacturing data analysis and update machine historical log.

8.3.1 Obtain Component Feature Fact File

The component feature fact file corresponds to the comparative tolerance analysis fact file created by the comparative tolerance analysis phase of PEPSIN from the data residing in the order sub-model (see chapter 9.3). The file contains all the necessary feature attributes and operational data in an object-oriented form. No other data regarding the condition of status of the component is requires to be supplied by the user during the consultation. This feature fact file is loaded into PADDES’s short term or working memory by the user.

The feature fact file in the short-term memory consists of feature class and instance definition statements. These statements require execution to create the classes and feature instances that represent the component’s status within the working memory. This is achieved though the CLIPS reset command, which establishes the component’s representation in the working memory in the appropriate form ready for consultation, as in figure 8.9.

8.3.2 Obtain Machine Dependent Fault Library

The activity is concerned with the input of the diagnostic decision logic expressed by the complete collection of decision trees for both individual and combined feature error diagnosis and implemented in PADDES as diagnostic rules. The diagnostic rules are obtained for the machine tool in question form the manufacturing sub-model (see chapter 9.4). These rules are loaded in PADDES’s long-term memory and constitute the domain knowledge contained within the knowledge base. Upon loading of the knowledge base the rules are examined by CLIPS to establish which rules will fire and
Figure 8.8 IDEF0 Representation of the Manufacturing Data Analysis Functionality of the PDA Framework

(a) Manufacturing Data Analysis, (b) Conduct MDA,
(b) Individual Feature MDA, (d) Combined Feature MDA
Figure 8.9 PADDES Expert System Environment with Feature Instances and Diagnostic Knowledge Base Loaded
(a) CLIPS Main Window, (b) CLIPS Agenda Window, (c) CLIPS (COOL) Instance Window,
(d) CLIPS Asserted Fact Window, (e) CLIPS Global Variable Window
these are placed in order of salience on an agenda, figure 8.9b, for execution.

8.3.3 Conduct Manufacturing Data Analysis

This activity of PADDES is initiated once both the component feature instances have been defined as shown in figure 8.9c and the knowledge base has been populated with the machine dependent diagnostic rules. The consultation is achieved by running the initial rules that appear within the agenda list, figure 8.9b. As the rules fire, new facts are asserted and retracted in the fact window, figure 8.9d, and new rules are activated and placed on the agenda when the conditional elements are satisfied. This procedure continues until all the rules that have been satisfied have fired thus ending the consultation.

As previously mentioned before in section 8.2.5, the logic followed by the execution of the consultation consists of individual feature manufacturing data analysis and combined feature manufacturing data analysis. The complete logic decision tree representation of both individual and combined manufacturing data analysis is contained in appendix III. The individual feature manufacturing data analysis is conducted for every feature attribute for each feature in the order of priority dictated by the salience value. Any production error, cause and suggested corrective action conclusions are appended to the PADDES analysis file during the consultation giving a running record of the analysis.

Combined feature manufacturing data analysis is initiated upon completion of the individual feature analysis and is solely concerned with the combined feature analysis of positional errors. This analysis checks all feature positional statuses in the X axis, then the Y axis, the combinatory effect of both the X and Y axes and finally the Z axis. The X and Y axes analysis is primarily concerned with the checking of positional errors to try and infer machine out-of-calibration or component set-up error form the individual feature positional errors. The Z axis analysis examines all the asserted Z axis facts of the features inserted directly onto the top face of the billet in an attempt to identify component set-up and billet size anomalies. If a conclusion can be drawn form this activity the original feature position conclusions are retracted and a new combined feature conclusion is asserted. This combined conclusion is also appended into the machine dependent historical log of the manufacturing sub-model giving an up-to-date record of the machine’s performance.
Although the activities involved in the PEPSIM phase of the novel PDA framework are conducted completely transparently to the user, PADDES employs a forward chaining inference strategy that requires no input on behalf of the user, the execution of the PADDES phase does however require a basic knowledge of the operational procedure of the CLIPS expert system shell.

8.4 Summary of Manufacturing Data Analysis and Diagnosis
The PADDES phase of the prototype PDA system employs the C Language Integrated Production System (CLIPS) version 6.1 expert system shell to provide the basic inference strategy. The inference strategy of CLIPS is of the forward chaining approach and is best suited to the analysis of previously defined facts, which requires little or no interaction on behalf of the consultant. PADDES utilised as its predefined facts an object–oriented comparative tolerance analysis fact file generated by the PEPSIM phase of the prototype PDA system.

The knowledge representation employed as the knowledge base of PADDES comprises feature-based rules that interrogate the component’s feature information contained within the comparative tolerance analysis fact file. This is conducted in an attempt to infer production errors, assign probable causes and suggest corrective action for milling, drilling, boring and reaming type operations that can be executed on a 3-axis machining centre.

The testing of the inspection methodology of PEPSIM and the logic employed during a consultation of PADDES of the prototype PDA system for a complex 2 ½ D prismatic component is detailed in chapter 10.
Chapter 9

INFORMATION MODELS TO SUPPORT PRODUCTION DATA ANALYSIS

9.1 Introduction
This chapter introduces the information requirements to support the functionality of the PDA framework’s PEPSIM and PADDES activities. The information requirement extends the work of the order information representation of the SME reference model proposed by Toh (1997, Toh et al. 1998) and facility manufacturing resource information representation introduced by Molina (1995).

The first section introduces the sub-model representation of the SME reference model followed by a description of the order sub-model structure and the order and customer information captured within it. The nominal and actual component extensions to the initial order sub-model to support the PEPSIM functionality of the PDA framework are outlined in detail.

The final section of this chapter outlines the representation of a highly automated manufacturing facility proposed by Molina (1995) and identifies the machine dependent information required to assist in the diagnostic execution of the PADDES phase of the PDA framework.

Although the research outlined in this thesis is underpinned by the information contained within information models and extends the work conducted by Toh (1997, Toh et al. 1998) and Molina (1995, Molina and Bell 1999), the identification and construction of these models is detailed within the aforementioned references and therefore is considered out of the scope of this research contribution by the author.

9.2 The SME Reference Model
As previously introduced in section 6.2.2. Toh (1997) developed the SME reference model as part of the EPSRC GR/L27077 project to establish the most appropriate “IT tool to Improve the Manufacturing Performance of Metalworking SMEs”. This reference model that was designed to represent the holonic characteristics of a contemporary metalworking SME and has been created from the interrelated three fields
of data capture namely: organisation/behaviour, facility and information fields of the SME enterprise modelling process. The first field of data capture describes the organisation and behaviour, or holonic characteristics, of the enterprise. The second field of data involves the description of the information attributes relating to the order, resources and process data. The third field of data represents the manufacturing facility description. These three fields of data capture are represented by three sub-models of the SME reference model:

i) the organisation-behaviour sub-model;
ii) the order sub-model; and
iii) the manufacturing sub-model.

The work reported in this thesis significantly extends the information representation captured by both the order sub-model and the manufacturing sub-model the description of which constitutes the remainder of the chapter.

9.3 The SME Order Sub-Model
Toh (1997) influenced by work conducted on order priority management by Nicholson (1985) and Westbrook (1993) recognised the requirement for the inclusion of order related information to be included within the SME reference model. The order sub-model plays an important role in the processing of fluctuating customer demands by enabling the processing of jobs with a minimal amount of planning and allowing job information to be accumulated as manufacturing processes. This order sub-model, depicted by the Booch (1994) class diagram in figure 9.1, defines the core information relating to the progress of work and identifies the status and location of orders and contains generic information that relates to the orders and jobs processed within the factory.

The information structure of the order sub-model proposed by Toh (1997, Toh et al 1998) is designed such that each order comprises a list of one or a number of jobs. The generic order and job objects of the Booch class diagram contains attributes that describes the scheduling and administration information which caters for different users within the factory. The order structure captures information for administration purposes such as: order number, order description, order price, order quantity, order delivery date and customer reference etc. The job structure on the other hand comprises
administration and scheduling type information such as: job number, job description, job quantity, job start date, job due date, job location and operation duration data such as set-up times, operation start and completion times. Job location information provides the ability to track the work-in-progress of jobs on the shop floor.

Figure 9.1 The Order Sub-model According to Toh (1997)

The order sub-model is directed at the contemporary metalworking SME that operates without a product design function and as such the order sub-model captures information relating to the product during the manufacturing phase of its life cycle. The information contained within the order sub-model comprises a subset of the data that would be included for some applications within a comprehensive product model (see chapter 3.5.1).
9.3.1 SME Order Sub-Model to Support Production Data Analysis

The PDA framework provides major extensions to the SME order sub-model. These extensions are semantically illustrated by the Booch diagram in figure 9.2 (Booch 1994) and constitute the addition of feature attribute information structures to augment the order and job information already captured within the model. This depicts two main categories of data held within the order model, namely: nominal feature attributes and actual feature attributes. These categories can be further subdivided into the following objects:

**Nominal Feature Attributes**

i) General attributes;  
ii) Nominal attributes;  
iii) Orientation attributes;  
iv) Tolerance specification;  
v) Datum references;  
vi) Probing points;  
vii) Operation attributes.

**Actual Feature Attributes**

i) Summary Information;  
ii) Tolerance Deviations;  
iii) Actual Attributes;  
iv) Actual Orientation.

9.3.1.1 Nominal Part Attribute Aspect of the PDA Framework's Order Sub-model

This portion of the PDA order sub-model captures the nominal or designed characteristics of the component. This data structure is populated with the nominal information transparently during the creation of the component model within PEPSIM. As this information forms the definitive representation of the component from which supports the entire functionality of the PDA framework only one representation of nominal part attributes is allowed within the order sub-model and hence the one-to-one cardinality. The remainder of this subsection is devoted to describing the nominal feature attributes captured within the nominal portion of the order sub-model.

i) **General attributes** can be categorised into general part attributes and general feature attributes. General part attributes capture the physical properties and characteristics of the component as a whole, whilst general feature attributes comprise the general feature identification attributes for each feature of the
component. The general attributes for both part and features are documented in table 9.1.
Nominal feature attributes represent the feature specific attributes that describe the desired dimensional characteristics of a particular feature. These attributes include: length, width, depth, diameter, counterbore diameter, counterbore length, corner radius and bottom radius that can be used to describe the feature geometry of any feature within the PEPSIM’s taxonomy. These attributes are by no means exhaustive and can be enhanced to support user defined features as required.

Orientation attributes comprise of the insertion point co-ordinates in the X, Y and Z axes of the feature and its associated rotation angle about each axis of orientation sensitive features such as key slot open, key slot closed, key slot plunged, closed blind rectangular pocket and open blind rectangular pocket.

Tolerance specification: All tolerances specified within ASME Y14.5M (ASME 1994) namely, size, form, profile, orientation, location and run-out, are catered for within this portion of the nominal feature object and includes tolerances for size, form, profile, orientation, location and run-out and are outlined in table 9.2.

<table>
<thead>
<tr>
<th>Tolerance Specification Attributes (Size)</th>
<th>Tolerance Specification Attributes (Form)</th>
<th>Tolerance Specification Attributes (Orientation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length Tolerance</td>
<td>Straightness Tolerance</td>
<td>Angularity Tolerance</td>
</tr>
<tr>
<td>Width Tolerance</td>
<td>Flatness Tolerance</td>
<td>Perpendicularity Tolerance</td>
</tr>
<tr>
<td>Depth Tolerance</td>
<td>Roundness Tolerance</td>
<td>Parallelism Tolerance</td>
</tr>
<tr>
<td>Diameter Tolerance</td>
<td>Cylindricity Tolerance</td>
<td></td>
</tr>
<tr>
<td>C/B Diameter Tolerance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C/B Length Tolerance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Corner Radius Tolerance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bottom Radius Tolerance</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tolerance Specification Attributes (Profile)</td>
<td>Tolerance Specification Attributes (Location)</td>
<td>Tolerance Specification Attributes (Run-out)</td>
</tr>
<tr>
<td>Profile of a Line Tolerance</td>
<td>Position X Tolerance</td>
<td>Circular Runout Tolerance</td>
</tr>
<tr>
<td>Profile of a Surface Tolerance</td>
<td>Position Y Tolerance</td>
<td>Total Runout Tolerance</td>
</tr>
<tr>
<td></td>
<td>Position Z Tolerance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Concentricity Tolerance</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Symmetry Tolerance</td>
<td></td>
</tr>
</tbody>
</table>

Table 9.2 Tolerance Attributes of the PDA Order Sub-model
v) **Operation data attributes** cover basic milling operations such as drilling, boring, milling and chamfering operation types and corresponds to the operation information employed within the NC part program. Table 9.3 provides a breakdown of the operation data attributes covered by the PDA order sub-model.

<table>
<thead>
<tr>
<th>Drilling/Reaming Operation Attributes</th>
<th>Boring Operation Attributes</th>
<th>Milling Operation Attributes</th>
<th>Chamfer Operation Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Operation Parameters</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Machining Stage</td>
<td>Machining Stage</td>
<td>Machining Stage</td>
<td>Machining Stage</td>
</tr>
<tr>
<td>Machining Sub-stage</td>
<td>Machining Sub-stage</td>
<td>Operation Type</td>
<td>Operation Type</td>
</tr>
<tr>
<td>Operation Type</td>
<td>Operation Sub-stage</td>
<td>Grouping Mode</td>
<td>Grouping Mode</td>
</tr>
<tr>
<td>Parameter List of Data</td>
<td>Parameter List of Data</td>
<td>Parameter List of Data</td>
<td>Parameter List of Data</td>
</tr>
<tr>
<td>Depth to Machine</td>
<td>Depth to Machine</td>
<td>Step-over</td>
<td>Step-over</td>
</tr>
<tr>
<td>Rapid to Depth</td>
<td>Rapid to Depth</td>
<td>Start Depth</td>
<td>Start Depth</td>
</tr>
<tr>
<td>Peck Depth</td>
<td>Peck Depth</td>
<td>Finish Depth</td>
<td>Finish Depth</td>
</tr>
<tr>
<td><strong>Operation Tool Data</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tool Type</td>
<td>Tool Type</td>
<td>Pre-machined Dia</td>
<td>Tool Type</td>
</tr>
<tr>
<td>Tool Lib. Search Mode</td>
<td>Tool Lib. Search Mode</td>
<td>Tool Left with</td>
<td>Tool Lib. Search Mode</td>
</tr>
<tr>
<td>Tool Major Dia.</td>
<td>Tool Major Dia.</td>
<td>Allowance</td>
<td>Tool Major Dia.</td>
</tr>
<tr>
<td>Tool Length</td>
<td>Tool Length</td>
<td>N/F Entry Distance</td>
<td>Tool Length</td>
</tr>
<tr>
<td>Tool Included Angle</td>
<td>Tool Included Angle</td>
<td>N/F Exit Distances</td>
<td>Tool Included Angle</td>
</tr>
<tr>
<td>Tool Bottom Dia.</td>
<td>Tool Bottom Dia.</td>
<td>X Entry</td>
<td>Tool Bottom Dia.</td>
</tr>
<tr>
<td>Spindle Speed</td>
<td>Spindle Speed</td>
<td>Y Entry</td>
<td>Spindle Speed</td>
</tr>
<tr>
<td>Vertical Feed</td>
<td>Vertical Feed</td>
<td>Entry Distance</td>
<td>Vertical Feed</td>
</tr>
<tr>
<td>Horizontal Feed</td>
<td>Horizontal Feed</td>
<td>Feed Type</td>
<td>Horizontal Feed</td>
</tr>
<tr>
<td>Z Peck Depth of Cut</td>
<td>Z Peck Depth of Cut</td>
<td>Bottom Radius</td>
<td>Z Peck Depth of Cut</td>
</tr>
<tr>
<td>Tool Teeth No.</td>
<td>Tool Teeth No.</td>
<td></td>
<td>Tool Teeth No.</td>
</tr>
<tr>
<td>Tool Material</td>
<td>Tool Material</td>
<td></td>
<td>Tool Material</td>
</tr>
<tr>
<td>Tool Description</td>
<td>Tool Description</td>
<td></td>
<td>Tool Description</td>
</tr>
<tr>
<td><strong>Operation Comment</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operation Description</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 9.3 Operation Data Attributes of the PDA Order Sub-model

Each operation type consists of three fields of data namely: operation tooling data, operation parameters and operation comment, which includes attributes
such as: tool geometry, tool material, spindle speeds, vertical and horizontal feeds and depth of cut information. The information is generated and transparently populated during the machine and inspection planning stage of PEPSIM. The tooling data forms the basis from which PADDES conducts its manufacturing data analysis diagnosis (see appendix III).

9.3.1.2 Actual Feature Attribute Aspect of the PDA Framework’s Order Sub-model

This portion of the PDA order sub-model constitutes the current state of the component and comprises of the calculated and analysed feature actuals that are generated through comparative tolerance analysis phase of PEPSIM. As this information is generated for every inspected component, the PDA order sub-model is capable of capturing multiple instances of actual part attributes that correspond to each inspected component. The rest of this subsection describes the actual feature attributes of the order sub-model that represents the current state of the inspected component.

i) Summary attributes: This object holds the general information regarding the actual measured component and includes data such as: part identification number, feature name and the overall quality status of the measured component.

ii) Tolerance Deviations capture the deviations as a result of the comparison of the nominal component geometry and the as-measured actual component geometry conducted by the comparative tolerance analysis phase of PEPSIM. It also includes feature tolerance status i.e. in-tolerance and out-of-tolerance, feature quality status of either accept, reject or rework. These attributes are also capable of supporting all of the size, form, profile, orientation, location, and run-out tolerance types stipulated within ANSI Y14.5M (ASME 1994).

iii) Actual feature attributes capture the same basic attribute information as the nominal object but with regard to the actual feature measured dimensions calculated by the initial phase of the comparative tolerance analysis stage of PEPSIM prior to the comparative analysis being conducted (see chapter 7.7). The actual portion of the order sub-model possesses the capability to capture sub-feature or geometric element details such as circle and plane details. The scope of actual feature attributes supported by the PDA order sub-model is documented in table 9.4.
### Actual Feature Attributes

- Actual Feature Length
- Actual Feature Width
- Actual Feature Depth
- Actual Feature Diameter
- Actual Feature C/B Diameter
- Actual Feature C/B Length
- Actual Feature Corner Radius
- Actual Feature Bottom Radius

### Actual Plane Sub-Feature Attributes

- Plane X Co-ordinates
- Plane Y Co-ordinates
- Plane Z Co-ordinates
- Plane Flatness

### Actual Circle Sub-Feature Attributes

- Centre X Co-ordinates
- Centre Y Co-ordinates
- Centre Z Co-ordinates
- Circle Roundness

**Table 9.4 Actual Feature Attributes of the PDA Order Sub-model**

iv) *Actual Orientation attributes* holds the feature transformation information in the form of actual insertion point co-ordinates in the X, Y, and Z axes. The feature orientation of rotation sensitive features is also captured although in the case of PEPSIM’s functionality this has been restricted to the rotation about the Z axis only.

### 9.4 The SME Manufacturing Sub-Model

The manufacturing information sub-model of the SME reference model provides the representation of the SME’s manufacturing facility, resources and its capabilities. The information contained within the manufacturing sub-model is employed in conjunction with the order sub-model for the management of order priorities within a contemporary SME. The manufacturing sub-model is based on research conducted by Molina (1995, 1999) for application within highly automated manufacturing systems and is structured around a four level characterisation of a manufacturing facility as depicted in Booch representation in figure 9.3.

![Figure 9.3 The Structure of the Manufacturing Sub-model (Molina 1995)](image)
The manufacturing sub-model describes the manufacturing facility at factory, shop, cell and workstation levels of abstraction. At each level the resources, processes and operational strategies can be defined to varying degrees of detail to support the diverse information requirements of the SME.

Work conducted by Toh et al (1998) has extended the manufacturing sub-model representation proposed by Molina (1995, Molina and Bell 1999) to capture information relating to the level and location of material, expendable inventory, durable operational inventory job sequences, and the impact of job relocation on the capacity loading at individual workstations. The manufacturing sub-model provides essential support for the identification of alternative process routes by identifying manufacturing processes and resource commitments of the factory at any particular time. The SME manufacturing sub-model proposed by Toh follows the original proposal of Molina with the exception of manufacturing strategies which were deemed inappropriate for an SME manufacturing environment.

9.4.1 SME Manufacturing Sub-Model to Support Production Data Analysis

The PDA framework extends the information representation captured by the resource portion of the manufacturing sub-model. The taxonomic information structure illustrated by the Booch class diagram of figure 9.4 captures all the information regarding the physical resources within the factory. These include resources such as: material handling resources, information processing resources, measurement and testing resources, human resources and production resources.

As represented by the Booch class diagram depicted in figure 9.4 the geometric errors identified by the PEPSIM phase of the PDA framework are produced by one or more of the production errors categorised in chapter 8.2.4. These production errors and their associated causes are dependent upon the individual machine tool responsible for the production of the geometric feature. As the Booch class resource taxonomy of the manufacturing sub-model suggests, every type of machine tool as well as every instance of that type within the manufacturing facility can be represented.

The PDA framework reported in this thesis influences the production resources portion of the manufacturing sub-model and extends the individual machine tool description at the leaves of the taxonomic structure to include a machine dependent fault library and historical log. Every individual machine tool possesses its unique fault library that represents its condition and machine capabilities.
The geometric errors identified through PEPSIM are mapped onto the machine dependent fault library in order to attempt to ascertain machine dependent production errors. A typical generic fault library entry for a positional error for a hole type construction feature is depicted in table 9.5 (Gillespie 1988). The fault library is represented in the form of diagnostic rules (see chapter 8) and can be specifically compiled to represent not only machines of differing configurations but also to represent individual machines of the same configuration. The logic employed within the machine dependent fault library is documented in appendix III. Although the machine dependent fault library of the manufacturing sub-model is generic in nature, and is directed toward a three axis vertical milling machine, it can be extended by the elicitation and introduction of local or operator experience knowledge of a specific machine.

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>Error Type</th>
<th>Geometric Fault</th>
<th>Possible Production Causes</th>
<th>Corrective Action / Feedback</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hole General</td>
<td>Positional</td>
<td>• Error of hole position in the X direction.</td>
<td>• Component set-up incorrect relative to fixture</td>
<td>• Check and reset component on fixture</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Error of hole position in the Y direction.</td>
<td>• Fixture set-up incorrect relative to machine datum</td>
<td>• Check and reset fixture relative to M/C Datum</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Error of hole position in both the X &amp; Y dir's</td>
<td>• Part programming error</td>
<td>• Check NC program movements with corresponding nominal feature positions</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Incorrect inaccuracies of machine tool</td>
<td></td>
<td>• Conduct a machine alignment test</td>
</tr>
</tbody>
</table>

Table 9.5 Hole Positional Error Representation of the Machine Dependent Fault Library (Adapted form Gillespie 1988)

The machine dependent historical log provides a running record of the machine type errors diagnosed for each machine and is generated as a consequence of conducting the manufacturing data analysis activity of PADDES. Its primary use is to aid the planner identifying the most appropriate machine to conduct the production activity and to indicate when preventative maintenance of the machine should be undertaken.
Figure 9.4 Representation of Manufacturing Resources within the PDA Manufacturing Sub-model (Adapted from Molina 1995, Molina et al 1999)
The product quality information held within order sub-model and the machine
diagnostic and performance information of the manufacturing sub-model consists of
information captured and generated from both PEPSIM and PADDES phases of the
PDA facility thus closing the manufacturing information loop that exists between the
available information resources of the contemporary metalworking SME. Therefore the
PDA framework’s extension to these models enhances the effectiveness of enterprise
information resources by integrating the information held within the order and
manufacturing sub-models so as to produce manufacturing performance information
from product quality feedback data.
Chapter 10

PRODUCTION DATA ANALYSIS TEST CASES

10.1 Introduction
The purpose of this chapter is to describe the issues relating to the design, implementation and analysis of results for a series of component related experiments to evaluate the functionality and efficiency of the PEPSIM and PADDES elements of the Production Data Analysis framework. The initial part of the chapter investigates the effectiveness of the individual phases involved within PEPSIM through the creation of a complex component part. The latter part of the chapter examines the manufacturing diagnostic logic employed within the knowledge base of PADDES based upon the comparative tolerance analysis fact file generated by the execution of PEPSIM.

10.2 Experimental Parts Design
The testing of the prototype PDA framework has been undertaken by employment of six component test pieces that represent the feature-based capabilities of the prototype system. These test cases have been applied to test the effectiveness of each of the phases of the PEPSIM and PADDES elements of the prototype framework. Although there have been a number of test case pieces adopted by many researchers to test the capabilities of feature-based modellers (Hounsell 1998) the author has deemed it necessary to design the test case components that best represent the feature taxonomy, inspection and diagnostic capabilities employed by the prototype PDA framework. The component configurations 1 to 6, employed to test the prototype system, are illustrated in figure 10.1 whilst a breakdown of their feature representations is documented in table 10.1. As the table 10.1 suggests all features contained within the feature taxonomy of the PDA framework have been utilised within the six component configurations with the exception of: drilled flat hole, reamed hole, bored hole and key slot plunged. This was considered to be adequate as the feature geometry of the omitted features is identical to that of other features employed in the construction of the component. Exemplars of this include: drilled flat hole and the bored hole of the socket head screw hole feature, bored / reamed holes and a drilled through hole feature and finally key slot
plunged and the key slot open features. The only difference between these pairings of features is in their method of manufacture.

Figure 10.1 PDA Framework Test Case Component Configurations
<table>
<thead>
<tr>
<th>Test 1</th>
<th>Test 2</th>
<th>Test 3</th>
<th>Test 4</th>
<th>Test 5</th>
<th>Test 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billet Dims</td>
<td>L 100</td>
<td>W 50</td>
<td>D 25</td>
<td>L 100</td>
<td>W 50</td>
</tr>
<tr>
<td>D Dia. 20 In: X 50 Y 25, Z - 10</td>
<td>D Dia. 20 In: X 50 Y 25, Z - 10</td>
<td>D Dia. 20 In: X 50 Y 25, Z - 10</td>
<td>D Dia. 20 In: X 50 Y 25, Z - 10</td>
<td>D Dia. 20 In: X 50 Y 25, Z - 10</td>
<td>D Dia. 20 In: X 50 Y 25, Z - 10</td>
</tr>
<tr>
<td>Drilled Through Hole</td>
<td>Drilled Through Hole</td>
<td>Drilled Through Hole</td>
<td>Drilled Through Hole</td>
<td>Drilled Through Hole</td>
<td>Drilled Through Hole</td>
</tr>
<tr>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
</tr>
<tr>
<td>Drilled Blind Hole</td>
<td>Drilled Blind Hole</td>
<td>Drilled Blind Hole</td>
<td>Drilled Blind Hole</td>
<td>Drilled Blind Hole</td>
<td>Drilled Blind Hole</td>
</tr>
<tr>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
<td>Dia. S x 15 In: X 75 Y 25, Z 0</td>
</tr>
<tr>
<td>Drilled Flat Hole</td>
<td>Drilled Flat Hole</td>
<td>Drilled Flat Hole</td>
<td>Drilled Flat Hole</td>
<td>Drilled Flat Hole</td>
<td>Drilled Flat Hole</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>Reamed Hole</td>
<td>Reamed Hole</td>
<td>Reamed Hole</td>
<td>Reamed Hole</td>
<td>Reamed Hole</td>
<td>Reamed Hole</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>Bored Hole</td>
<td>Bored Hole</td>
<td>Bored Hole</td>
<td>Bored Hole</td>
<td>Bored Hole</td>
<td>Bored Hole</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>Counter-bored Hole</td>
<td>Counter-bored Hole</td>
<td>Counter-bored Hole</td>
<td>Counter-bored Hole</td>
<td>Counter-bored Hole</td>
<td>Counter-bored Hole</td>
</tr>
<tr>
<td>Socket Head Hole</td>
<td>Socket Head Hole</td>
<td>Socket Head Hole</td>
<td>Socket Head Hole</td>
<td>Socket Head Hole</td>
<td>Socket Head Hole</td>
</tr>
<tr>
<td>H.Dia. 10.5, S.H.Dia 17, C.B.L. 10 In: X 50 Y 50, Z - 10</td>
<td>H.Dia. 10.5, S.H.Dia 17, C.B.L. 10 In: X 50 Y 50, Z - 10</td>
<td>H.Dia. 10.5, S.H.Dia 17, C.B.L. 10 In: X 50 Y 50, Z - 10</td>
<td>H.Dia. 10.5, S.H.Dia 17, C.B.L. 10 In: X 50 Y 50, Z - 10</td>
<td>H.Dia. 10.5, S.H.Dia 17, C.B.L. 10 In: X 50 Y 50, Z - 10</td>
<td>H.Dia. 10.5, S.H.Dia 17, C.B.L. 10 In: X 50 Y 50, Z - 10</td>
</tr>
<tr>
<td>Key Slot Closed</td>
<td>Key Slot Closed</td>
<td>Key Slot Closed</td>
<td>Key Slot Closed</td>
<td>Key Slot Closed</td>
<td>Key Slot Closed</td>
</tr>
<tr>
<td>Key Slot Open</td>
<td>Key Slot Open</td>
<td>Key Slot Open</td>
<td>Key Slot Open</td>
<td>Key Slot Open</td>
<td>Key Slot Open</td>
</tr>
<tr>
<td>L 20, W 15, D 10 In: X 100 Y 25, Z 0 Rot @ Z: 90 deg</td>
<td>L 20, W 15, D 10 In: X 100 Y 25, Z 0 Rot @ Z: 90 deg</td>
<td>L 20, W 15, D 10 In: X 100 Y 25, Z 0 Rot @ Z: 90 deg</td>
<td>L 20, W 15, D 10 In: X 100 Y 25, Z 0 Rot @ Z: 90 deg</td>
<td>L 20, W 15, D 10 In: X 100 Y 25, Z 0 Rot @ Z: 90 deg</td>
<td>L 20, W 15, D 10 In: X 100 Y 25, Z 0 Rot @ Z: 90 deg</td>
</tr>
<tr>
<td>Key Slot Plunged</td>
<td>Key Slot Plunged</td>
<td>Key Slot Plunged</td>
<td>Key Slot Plunged</td>
<td>Key Slot Plunged</td>
<td>Key Slot Plunged</td>
</tr>
<tr>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>L 30, W 30, D 10 CR 5 In: X 50 Y 25, Z 0 Rot @ Z: 180 deg</td>
<td>L 30, W 30, D 10 CR 5 In: X 50 Y 25, Z 0 Rot @ Z: 180 deg</td>
<td>L 30, W 30, D 10 CR 5 In: X 50 Y 25, Z 0 Rot @ Z: 180 deg</td>
<td>L 30, W 30, D 10 CR 5 In: X 50 Y 25, Z 0 Rot @ Z: 180 deg</td>
<td>L 30, W 30, D 10 CR 5 In: X 50 Y 25, Z 0 Rot @ Z: 180 deg</td>
<td>L 30, W 30, D 10 CR 5 In: X 50 Y 25, Z 0 Rot @ Z: 180 deg</td>
</tr>
<tr>
<td>Round Blind Pocket</td>
<td>Round Blind Pocket</td>
<td>Round Blind Pocket</td>
<td>Round Blind Pocket</td>
<td>Round Blind Pocket</td>
<td>Round Blind Pocket</td>
</tr>
<tr>
<td>Dia. 40, D 10 In: X 50 Y 25, Z 0</td>
<td>Dia. 40, D 10 In: X 50 Y 25, Z 0</td>
<td>Dia. 40, D 10 In: X 50 Y 25, Z 0</td>
<td>Dia. 40, D 10 In: X 50 Y 25, Z 0</td>
<td>Dia. 40, D 10 In: X 50 Y 25, Z 0</td>
<td>Dia. 40, D 10 In: X 50 Y 25, Z 0</td>
</tr>
</tbody>
</table>

Table 10.1 Test Case Component Feature Specification
10.3 Production Data Analysis Component Study

Due to the complexity of the functionality of the prototype PDA system the experimental testing documented in this chapter will be directed at test component number 6.

This section proves the methodology of the PDA concept and is based on component 6, which has been selected due to it possessing both the greatest degree of complexity and also is constructed using the widest variety of manufacturing features. The following sub-sections outline the six operational phases of the prototype PDA system, as depicted within figure 10.2, the users decision making process whilst describing the information requirements of each phase.

The following sub-sections describe the processes involved in the creation of component geometry, production code generation and subsequent analysis of test component number 6 whilst illustrating the documentation produced through the execution of both the PEPSIM and PADDES elements of the PDA framework (see figure 6.3). The logic representation adopted in the knowledge base of the PADDES diagnostic expert system will be extensively tested through the application of simulated production errors produced on a CNC three-axis vertical machining centre.

The overall PDA framework described in chapter 9 and illustrated in figure 10.2 consist of six major phases that constitute the PEPSIM and PADDES elements of the prototype PDA system and comprise of the following:

i) *Create part model geometry* of PEPSIM (figure 10.2a) involves the initial creation of the component's billet representation and subtracting the individual depression type construction feature geometry and the associated tolerance specification that collectively constitutes the component's geometry from the billet's positive geometry.

ii) *Create operation plan* of PEPSIM (figure 10.2b) involves the identification of specific operations, their sequence and associated tooling required to produce a construction feature. The completed operation plan entails the sequencing of all component operations to minimise tool changes.

iii) *Create inspection procedure* of PEPSIM (figure 10.2c) entails the generation of an inspection plan that involves the algorithmic generation of probing points and probing paths for each feature and the subsequent joining of these feature-based probing paths together to form the complete inspection plan for the component.
Figure 10.2 Stages of the Experimental Design and Evaluation of the PDA Framework
As the methodology employed within the PEPSIM element of the PDA framework utilises a single vertical tactile probe orientation the sequencing of inspection has been restricted to the order of construction feature creation although the concept of feature priority is supported within the prototype PDA system.

iv) *Produce production code* of PEPSIM (figure 10.2d) involves the transparent and simultaneous generation of tooling set-up sheets and machine dependent NC part programs from the post processing cutter location data and the operation plan and the feature-based macro generation of the CMM dependent inspection programs from the probing point and path data.

v) *Comparative tolerance analysis* is the final phase of PEPSIM (figure 10.2e) and entails the evaluation of the results obtained through the automatic execution of the CMM dependent inspection program generated through the latter phase of PEPSIM. The construction feature actuals are calculated and compared with the feature nominals in order to obtain the associated deviations. The feature deviations are subsequently compared against the tolerance specification specified during the construction of the feature to ascertain its tolerance condition and quality status.

vi) *Manufacturing data analysis* is conducted by the PADDES diagnostic expert system element of the prototype PDA system (figure 10.2f) and is concerned with the diagnosis of machine based production errors. PADDES conducts its diagnosis from the feature nominal attribute data, feature actual attribute data, feature status data, nominal feature operation data generated form the PEPSIN element of the PDA framework (figure 10.2a – 10.2e).

The remainder of this chapter is concerned with the testing of the aforementioned phases of the prototype PDA systems through the application of these phases to the construction, manufacture, inspection, comparative tolerance analysis and manufacturing data analysis of test component number 6. Each phase is accompanied by a detailed description of the user interactions, data requirements, production program structure, comparative analysis procedure, manufacturing data analysis logic and associated documentation involved in the prototype PDA system.
10.3.1 Create Part Model Geometry

The most labour intensive phase for the user of the prototype PDA system is the initial construction of the component’s feature geometry, illustrated in figure 10.2a. This phase involves the initial specification of the rectangular billet stock material that forms the canvas onto which the depression type construction features are inserted to create the overall feature geometry of the final component. The creation of a rectangular billet feature of test component number 6 is illustrated by the dialogue interaction depicted in figure 10.3a.

(a) Test Component Number 6 Billet Feature Creation Procedure

<table>
<thead>
<tr>
<th>FEAT_NAME</th>
<th>FEAT_TYPE</th>
<th>VDM_NO</th>
<th>FEAT_PP_NO</th>
<th>INS_PNTX</th>
<th>INS_PNTY</th>
<th>INS_PNTZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>BILLET</td>
<td>RECT</td>
<td>4200</td>
<td>12</td>
<td>50.00</td>
<td>50.00</td>
<td>0.00</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEAT_ROTX</th>
<th>FEAT_ROTY</th>
<th>FEAT_ROTZ</th>
<th>FEAT_LGH</th>
<th>FEAT_WTH</th>
<th>FEAT_DTH</th>
<th>FLAT_TOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>100.00</td>
<td>100.00</td>
<td>25.00</td>
<td>0.100</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P_PNTX1</th>
<th>P_PNTY1</th>
<th>P_PNTZ1</th>
<th>P_PNTX2</th>
<th>P_PNTY2</th>
<th>P_PNTZ2</th>
<th>P_PNTX3</th>
<th>P_PNTY3</th>
<th>P_PNTZ3</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0000</td>
<td>90.0000</td>
<td>0.0000</td>
<td>90.0000</td>
<td>90.0000</td>
<td>0.0000</td>
<td>90.0000</td>
<td>10.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P_PNTX4</th>
<th>P_PNTY4</th>
<th>P_PNTZ4</th>
<th>P_PNTX5</th>
<th>P_PNTY5</th>
<th>P_PNTZ5</th>
<th>P_PNTX6</th>
<th>P_PNTY6</th>
<th>P_PNTZ6</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0000</td>
<td>10.0000</td>
<td>0.0000</td>
<td>10.0000</td>
<td>0.0000</td>
<td>-5.0000</td>
<td>90.0000</td>
<td>0.0000</td>
<td>-5.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P_PNTX7</th>
<th>P_PNTY7</th>
<th>P_PNTZ7</th>
<th>P_PNTX8</th>
<th>P_PNTY8</th>
<th>P_PNTZ8</th>
<th>P_PNTX9</th>
<th>P_PNTY9</th>
<th>P_PNTZ9</th>
</tr>
</thead>
<tbody>
<tr>
<td>90.0000</td>
<td>0.0000</td>
<td>-10.0000</td>
<td>10.0000</td>
<td>0.0000</td>
<td>-10.0000</td>
<td>0.0000</td>
<td>10.0000</td>
<td>-5.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>P_PNTX10</th>
<th>P_PNTY10</th>
<th>P_PNTZ10</th>
<th>P_PNTX11</th>
<th>P_PNTY11</th>
<th>P_PNTZ11</th>
<th>P_PNTX12</th>
<th>P_PNTY12</th>
<th>P_PNTZ12</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>90.0000</td>
<td>-5.0000</td>
<td>20.0000</td>
<td>15.0000</td>
<td>10.0000</td>
<td>0.0000</td>
<td>10.0000</td>
<td>-10.0000</td>
</tr>
</tbody>
</table>

(b) Billet Feature Nominal Attribute Representation within the PDA Order Sub-model

Figure 10.3 PEPSIM’s Billet Feature Creation and Order Model Representation
This user interaction involves the specification of the overall dimensions of the billet, the origin co-ordinates i.e. the top left-hand corner of the billet, material type and properties and the inspection mode of either no inspection, all feature inspection or critical feature inspection. At present the scope of the prototype PDA system does not support dimensional tolerance information regarding the size of the original billet, however the flatness tolerance of the prototype system is defaulted at 0.1 mm.

The nominal dimension attributes are extracted during feature construction and transparently employed to instantiate the nominal portion of the PDA order sub-model which is captured in the form of a nominal attribute relational database embedded within the PDA framework the representation of which is illustrated in figure 10.3b. These captured billet nominal information attributes include: feature name, feature type, probing point number, billet insertion co-ordinates, billet rotational orientation, billet size attributes and probing point co-ordinates.

A similar procedure is followed during the addition of the depression type construction features to the billet feature. The feature construction procedure for the closed blind rectangular pocket and a related counterbored hole type features of test component number 6 are depicted in figure 10.4. The PDA order sub-model nominal representation for both the aforementioned construction features are documented in figure 10.5.
<table>
<thead>
<tr>
<th>FEAT_NAME</th>
<th>FEAT_TYPE</th>
<th>FEAT_NO</th>
<th>PRIORITY</th>
<th>VSM_NO</th>
<th>FEAT_PP_NO</th>
<th>INS_PNTX</th>
<th>INS_PNTY</th>
<th>INS_PNTZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK_000</td>
<td>PCB</td>
<td>1</td>
<td>1</td>
<td>4201</td>
<td>32</td>
<td>50.0000</td>
<td>50.0000</td>
<td>0.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEAT_ROTX</th>
<th>FEAT_ROTY</th>
<th>FEAT_ROTX</th>
<th>FEAT_LGH</th>
<th>FEAT_WTH</th>
<th>FEAT_DTH</th>
<th>FEAT_DIA</th>
<th>FEAT_CNDIA</th>
<th>FEAT_CRLGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>45.0000</td>
<td>40.0000</td>
<td>40.0000</td>
<td>10.0000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEAT_CRAD</th>
<th>FEAT_BRAD</th>
<th>DTM_FREAT</th>
<th>BILL_REF</th>
<th>LGH_TOL</th>
<th>WITH_TOL</th>
<th>DTH_TOL</th>
<th>DIA_TOL</th>
<th>CRLDH_TOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.0000</td>
<td>0.0000</td>
<td>None</td>
<td>0.100</td>
<td>0.100</td>
<td>0.100</td>
<td>0.100</td>
<td>0.100</td>
<td>0.100</td>
</tr>
</tbody>
</table>

(a) Closed Blind Rectangular Pocket Feature Nominal Attribute Representation within the PDA Order Sub-model

<table>
<thead>
<tr>
<th>FEAT_NAME</th>
<th>FEAT_TYPE</th>
<th>FEAT_NO</th>
<th>PRIORITY</th>
<th>VSM_NO</th>
<th>FEAT_PP_NO</th>
<th>INS_PNTX</th>
<th>INS_PNTY</th>
<th>INS_PNTZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>CB_000</td>
<td>HCB</td>
<td>2</td>
<td>1</td>
<td>402</td>
<td>20</td>
<td>50.0000</td>
<td>50.0000</td>
<td>-10.0000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEAT_ROTX</th>
<th>FEAT_ROTY</th>
<th>FEAT_ROTX</th>
<th>FEAT_LGH</th>
<th>FEAT_WTH</th>
<th>FEAT_DTH</th>
<th>FEAT_DIA</th>
<th>FEAT_CNDIA</th>
<th>FEAT_CRLGH</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>0.0000</td>
<td>15.0000</td>
<td>16.0000</td>
<td>24.0000</td>
<td>8.0000</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>FEAT_CRAD</th>
<th>FEAT_BRAD</th>
<th>DTM_FREAT</th>
<th>BILL_REF</th>
<th>LGH_TOL</th>
<th>WITH_TOL</th>
<th>DTH_TOL</th>
<th>DIA_TOL</th>
<th>CRLDH_TOL</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK_000</td>
<td>0.100</td>
<td></td>
<td>0.100</td>
<td>0.100</td>
<td>0.100</td>
<td>0.100</td>
<td>0.100</td>
<td>0.100</td>
</tr>
</tbody>
</table>

(b) Counterbored Hole Feature Nominal Attribute Representation within the PDA Order Sub-model

Figure 10.5 Typical PDA Order Sub-model Representation of the PEPSIM’s Construction Features
The nominal construction feature attributes are captured transparently by the PDA order sub-model during feature creation and comprise: feature name and type, feature number, feature priority, probing point number, insertion co-ordinates, orientation angle, feature dimensions, feature tolerances and the co-ordinates of the algorithmically generated probing points. Although the nominal portion of the PDA order sub-model is capable of capturing all the tolerance types stipulated within ASME Y14.5M, the PEPSIM phase of the prototype PDA system’s supported tolerances are restricted to those of dimensional, form, location and orientation types as documented in table 7.4. For the purposes of testing the PEPSIM phase of the prototype PDA system the tolerance values have been defaulted at ± 0.1 mm for dimensional and location type tolerances and ± 0.05 mm for form and orientation tolerance types.

The DTM_FEAT attribute captures whether the feature is inserted on the top surface of the billet as in the case illustrated in figure 10.5a or in the base of another feature depicted in figure 10.5b. The BILL_REF attribute is applied to open type construction features and captures the side reference of the billet upon which the open feature is inserted. Each construction feature comprises the same attribute structure and possesses its own record within the order sub-model relational database.

10.3.2 Create Operation Plan
The activity remains unchanged from the functionality of the original PEPS milling expert CAM software package, which forms the foundations of the prototype PDA system. The tool selection and operation parameter determination are achieved upon feature creation and follow the logic documented within appendix I. Each construction feature added to the component possesses operation attributes generated through rule-based macros, which include operation comment, operation tool/cut data and operation data. The determined operation comment, operation tool/cut data and operation data for the closed blind rectangular pocket and the related counterbored hole type construction feature employed as part of the feature description of test component number 6 are illustrated in figure 10.6. The operation comment element of the operation plan consists of the basic description of the operation type or cut type of either roughing, semi-finishing or finishing, centre drill, drill etc.
Figure 10.6 Feature-based Operation Plan Determination of PEPSIM Phase of the Prototype PDA Framework

The operation plan for the component as a whole for the given optimisation mode can be previewed at any time by the planner and comprises a complete ordered operation list and is based upon minimum number of tool changes. The complete ordered operation list for test component number 6 is illustrated in figure 10.7. This operation list includes all the tools required, their magazine station numbers and the associated cutting parameters. By analysing this ordered list the planner can identify whether any operation modifications are required. This complete operation plan can be visualised on the graphic user interface of the PEPS CAM software package. A selection of visualised feature operations for the closed blind rectangular pocket and counterbored construction features together with their associated operations is depicted in figure 10.7.

It must be pointed out that the information employed in the determination of the operation information is at this point in time contained within the data sets of an embedded tool library of the PEPS milling expert CAM software package. The PDA order sub-model is not populated with the feature-based operation information until the completion of the comparative tolerance analysis element of the PEPSIM phase of the prototype PDA systems.
10.3.3 Create Inspection Procedure

This activity represents one of the major and novel enhancements of the PEPS milling expert CAM software package to realise the PEPSIM phase of the prototype PDA system, illustrated in figure 10.2c.

Figure 10.7 Sequenced Operation Plan and Plan Simulation of Test Component Number 6
This activity is conducted simultaneously and transparently to the planner to the operation planning activity of PEPSIM and involves:

i) inspection planning;

ii) algorithmic feature-based probing point and path generation;

iii) inspection path simulation.

The inspection planning stage is primarily concerned with the identification of the features that require inspection and their sequence. As previously mentioned in chapter 7 the scope of the inspection strategy adopted by PEPSIM has been restricted to a single vertical probe orientation. This assumption negates the requirement for inspection feature sequencing with respect to minimising the number of probe orientation changes, as each depression feature supported by the prototype PDA system can be accessed with a single vertical probe configuration. Therefore the order in which the features are inspected follows the feature creation sequence.

All the features that comprise the overall geometry of the component can be inspected by specifying the inspection mode of all features during the specification of the part layout and billet details as in figure 10.3a. Alternatively selected features need only be inspected by specifying the individual feature priority during feature creation and selecting the inspection mode of critical features only at the creation of the part layout. Figure 10.8 illustrates the effect of applying both modes of inspection to test component number 6. All feature inspection is indicated by the inspection plan (a) whilst the inspection plan (b) documents the effect of specifying that only the closed blind rectangular pocket requires inspection.

During the creation and insertion of the construction features onto the billet’s top surface feature-based algorithms contained within macros of PEPSIM transparently generate the uniform probing point distribution which is employed during the inspection of the feature. The three dimensional probing point distributions follows the methodology specified previously in chapter 7.3.2 and is in accordance to BS7172. The uniform probing point distribution for each construction feature for test component number 6 is graphically depicted in figure 10.9.

The feature-based probing path for each feature is created by assigning approach and retract distances for each probing point of the feature. These approach/retract
**TOTAL FEATURE INSPECTION**

**Inspection Plan TEST6.tif**

- **Date**: Thursday 19th August 1999  
- **Time**: 10:40:11

**Summary Information:**

<table>
<thead>
<tr>
<th>Part Number</th>
<th>TEST6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part File Name</td>
<td>TEST6 MIL</td>
</tr>
<tr>
<td>Part NC Tape File Name</td>
<td>TEST6 TAP</td>
</tr>
<tr>
<td>Part HTBasic File Name</td>
<td>TEST6 X</td>
</tr>
<tr>
<td>Tip Calibration File Name</td>
<td>TEST6.C</td>
</tr>
<tr>
<td>Inspection Mode</td>
<td>All Feature Inspection</td>
</tr>
</tbody>
</table>

**Billet Details:**

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>RECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billlet Length (X)</td>
<td>100</td>
</tr>
<tr>
<td>Billlet Width (Y)</td>
<td>100</td>
</tr>
<tr>
<td>Billlet Depth (Z)</td>
<td>25</td>
</tr>
</tbody>
</table>

**PK_000 Feature Details:**

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>PCBR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCBR Pocket Length</td>
<td>40</td>
</tr>
<tr>
<td>PCBR Pocket Width</td>
<td>40</td>
</tr>
<tr>
<td>PCBR Pocket Depth</td>
<td>10</td>
</tr>
<tr>
<td>PCBR Pocket Corner Radius</td>
<td>5</td>
</tr>
<tr>
<td>PCBR Pocket Bottom Radius</td>
<td>0</td>
</tr>
<tr>
<td>PCBR Insertion Point (X)</td>
<td>50</td>
</tr>
<tr>
<td>PCBR Insertion Point (Y)</td>
<td>50</td>
</tr>
<tr>
<td>PCBR Insertion Point (Z)</td>
<td>0</td>
</tr>
<tr>
<td>PCBR Rotation About X Axis</td>
<td>0</td>
</tr>
<tr>
<td>PCBR Rotation About Y Axis</td>
<td>0</td>
</tr>
<tr>
<td>PCBR Rotation About Z Axis</td>
<td>45</td>
</tr>
<tr>
<td>Feature Number</td>
<td>1</td>
</tr>
<tr>
<td>Feature Priority</td>
<td>Critical</td>
</tr>
<tr>
<td>Number of Probing Points</td>
<td>32</td>
</tr>
</tbody>
</table>

**CB_000 Feature Details:**

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>HCRR</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCRR Hole Diameter</td>
<td>16</td>
</tr>
<tr>
<td>HCRR Hole Depth</td>
<td>15</td>
</tr>
<tr>
<td>HCRR Countersink Diameter</td>
<td>24</td>
</tr>
<tr>
<td>HCRR Countersink Depth</td>
<td>8</td>
</tr>
<tr>
<td>HCRR Insertion Point (X)</td>
<td>50</td>
</tr>
<tr>
<td>HCRR Insertion Point (Y)</td>
<td>50</td>
</tr>
<tr>
<td>HCRR Insertion Point (Z)</td>
<td>-10</td>
</tr>
<tr>
<td>Feature Number</td>
<td>2</td>
</tr>
<tr>
<td>Feature Priority</td>
<td>Critical</td>
</tr>
<tr>
<td>Number of Probing Points</td>
<td>20</td>
</tr>
</tbody>
</table>

**K_000 Feature Details:**

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>KEYC</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEYC Key Slot Length</td>
<td>50</td>
</tr>
<tr>
<td>KEYC Key Slot Width</td>
<td>15</td>
</tr>
<tr>
<td>KEYC Key Slot Depth</td>
<td>15</td>
</tr>
<tr>
<td>KEYC Insertion Point (X)</td>
<td>50</td>
</tr>
</tbody>
</table>

---

**CRITICAL FEATURE INSPECTION**

**Inspection Plan TEST6.tif**

- **Date**: Sunday 29th August 1999  
- **Time**: 11:42:45

**Summary Information:**

<table>
<thead>
<tr>
<th>Part Number</th>
<th>TEST6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part File Name</td>
<td>TEST6 MIL</td>
</tr>
<tr>
<td>Part NC Tape File Name</td>
<td>TEST6 TAP</td>
</tr>
<tr>
<td>Part HTBasic File Name</td>
<td>TEST6 X</td>
</tr>
<tr>
<td>Tip Calibration File Name</td>
<td>TEST6.C</td>
</tr>
<tr>
<td>Inspection Mode</td>
<td>Critical Feature Inspection Only</td>
</tr>
</tbody>
</table>

**Billet Details:**

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>RECT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billlet Length (X)</td>
<td>100</td>
</tr>
<tr>
<td>Billlet Width (Y)</td>
<td>100</td>
</tr>
<tr>
<td>Billlet Depth (Z)</td>
<td>25</td>
</tr>
<tr>
<td>Billlet Insertion Point (X)</td>
<td>50</td>
</tr>
<tr>
<td>Billlet Insertion Point (Y)</td>
<td>50</td>
</tr>
<tr>
<td>Billlet Insertion Point (Z)</td>
<td>0</td>
</tr>
</tbody>
</table>

**PK_000 Feature Details:**

<table>
<thead>
<tr>
<th>Feature Type</th>
<th>PCBR</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCBR Pocket Length</td>
<td>40</td>
</tr>
<tr>
<td>PCBR Pocket Width</td>
<td>40</td>
</tr>
<tr>
<td>PCBR Pocket Depth</td>
<td>10</td>
</tr>
<tr>
<td>PCBR Pocket Corner Radius</td>
<td>5</td>
</tr>
<tr>
<td>PCBR Pocket Bottom Radius</td>
<td>0</td>
</tr>
<tr>
<td>PCBR Insertion Point (X)</td>
<td>50</td>
</tr>
<tr>
<td>PCBR Insertion Point (Y)</td>
<td>50</td>
</tr>
<tr>
<td>PCBR Insertion Point (Z)</td>
<td>0</td>
</tr>
<tr>
<td>PCBR Rotation About X Axis</td>
<td>0</td>
</tr>
<tr>
<td>PCBR Rotation About Y Axis</td>
<td>0</td>
</tr>
<tr>
<td>PCBR Rotation About Z Axis</td>
<td>45</td>
</tr>
<tr>
<td>Feature Number</td>
<td>1</td>
</tr>
<tr>
<td>Feature Priority</td>
<td>Critical</td>
</tr>
<tr>
<td>Number of Probing Points</td>
<td>32</td>
</tr>
</tbody>
</table>

---

Figure 10.8 PEPSIM’s Test Component Number 6 Inspection Plans

195
Figure 10.9 Algorithmic Feature-based Three Dimensional Probing Point Distribution of the Construction Features of Test Component Number 6

(a) Billet Feature, (b & c) Closed Blind Rectangular Pocket and Counterbored Hole, (d & e) Key Slot Closed, (f & g) Open Blind Rectangular Pocket and Drilled Hole Through, (h & i) Socket Head Screw Hole

points are for each basic geometric element that constitutes the feature geometry and are joined together to produce a complete probing path for that particular geometric element. The resultant geometric element probing paths are subsequently joined together in sequence with the feature entry and exit path to produce the complete probing path for the feature. The automatically generated feature-based probing paths for the feature of test component number 6 are illustrated in figure 10.9.

The individual feature probing paths are connected together by a movement path in a plane that is a safe distance above the top surface of the billet thus ensuring a collision free transition movement from one feature to the next. The order of feature inspection follows the sequence specified within the operation plan. The completed component inspection probing point path can be verified through simulation on the graphic user interface of PEPSIM. The author acknowledges that the probing path
produced by PEPSIM does not provide the most efficient movement route, however it
does provide a consistent and robust method of directing the touch probe during
inspection.

The nominal portion of the PDA order sub-model, as depicted in figures 10.3b
and 10.5, is automatically populated with the ordered probing point co-ordinates
generated during the creation and insertion of the feature onto the billet surface by the
planner.

10.3.4 Produce Production Code

This activity of PEPSIM is concerned with the simultaneous transformation of the
information generated by both the operation and inspection planning activities into
machine dependent NC part programs and inspection code for automatic execution on a
three axis vertical machining centre and a CMM respectively as shown in figure 10.2d.

The machine dependent NC part program generation within PEPSIM converts
the cutter location data and tooling data generated by the operation planning activity and
follows the methodology of the original PEPS milling expert CAM software package.
The transformation is achieved through the use of a user configurable post processor,
which transparently utilises the tooling and cutter location data to produce a tool set-up
sheet and the machine dependent NC part program, as depicted for test component
number 6 in figure 10.10.

---

**Tool Set-up Sheet (.set file)**

FROM Position: X0 Y0 Z200

*** JOB SIDE 1

------------------------------------------
Tool Number 11 Block at Tool Change: 60
Tool Type: SLOT DRILL M7 D20 * 20
Tool Full Diameter: 20.0
Tool Length: 20.0
PK: 000 ROUGH S/S
Spindle Speed: 509 rpm 32.0 m/min
Feed Horizontal: 9.0 mm/min, 0.089 mm/tooth
Feed Vertical: 2.4 mm/min, 0.022 mm/tooth

------------------------------------------
Tool Number 8 Block at Tool Change: 250
Tool Type: NEW SLOT DRILL M7 D8 * 13
Tool Full Diameter: 8.0
Tool Length: 13.0
PK: 000 FINISH AROUND
Spindle Speed: 1273 rpm 32.0 m/min
Feed Horizontal: 6.3 6/mm/min, 0.025 mm/tooth
Feed Vertical: 20.3 6/mm/min, 0.008 mm/tooth

---

**Wadkin 4/6 Vertical Machining Centre
N.C. Part Program (.tap file)**

N05 (ID:PROG,785001,TEST6A)
N06 G71 G90 G43 G80
N10 (*2 JOB SIDE 1)
N20 (TOOL 11 SLOT DRILL M7 D20 * 20)
N30 (OFFSET LENGTH 11 RADIUS 11 )
N40 T11 M6
N50 D11 E10
N60 (PK: 000 ROUGH S/S)
N70 S509 M3
N80 G6 X50.0 Y90.0
N90 Z3.0
N100 G1 Z-10.0 F60.0
N110 X56.718 Y43.282 Z90.6
N120 X63.435 Y50.0
N130 X50.0 Y63.435
N140 X36.565 Y50.0
N150 X43.282 Y43.282
N160 X50.0 Y36.565
N170 X56.718 Y43.282
N180 Z-8.0 F60.0
N190 G6 Z120.0
N200 M9
N210 (TOOL 8 NEW SLOT DRILL M7 D8 * 13)
N220 (OFFSET LENGTH 8 RADIUS 8 )
N230 T8 M6
N240 D8 E10
N250 (PK: 000 FINISH AROUND)
N260 S1273 M3
N270 (G6 X9.393 Y46.464)
N280 G1 G41 D60 X39.04 Y46.111 F63.7

---

Figure 10.10 Tool Set-up Sheet and NC Part Program
Fragment for Test Component Number 6
The tool set-up sheet provides a comprehensive list of the tooling required to during the execution of the NC part program. This information includes for each tool: tool type, tool dimensions, tool magazine location, operation type and operation parameters such as speeds and horizontal and vertical feeds. The NC part program provides the data required by the vertical machining centre for axis servo control and ancillary functions.

The generation of the inspection plan, depicted in figure 10.8, and the CMM dependent inspection code within PEPSIM is achieved in a different manner to that of the tool set-up sheet and NC part program generation. PEPSIM employs a feature-based approach to generate both the operation plan and the CMM inspection code. As previously mentioned the inspection plan contains an ordered list of the features and their associated nominal attributes involved in the inspection of the component. The inspection program for the CMM is generated by feature-based macros in the PEPSIM phase of the prototype PDA system and at present can be in either one of two formats: the DMIS or the HTBasic native CMM format. The structure of the CMM inspection program generated by PEPSIM is independent on the format chosen and comprises the following elements as illustrated in figure 10.11:

i) CMM initialisation;  iv) Measure billet faces;
ii) Probe tip calibration;  v) Measure component feature;
iii) Component alignment;  vi) Output actual feature details.

Probe tip calibration and the basic six point component alignment are achieved manually by the CMM operator, whilst the measurement of the billet and subsequent features is conducted under servo control once the component alignment has been completed.

All the information required in the generation of the complete inspection plan and CMM inspection program is transparently extracted form the nominal portion of the order sub-model. This allows the powerful capability of automatic generation of executable CMM inspection programs to be achieved, simultaneously with the NC part program code, without any interaction or prior inspection programming knowledge on behalf of the planner.
Figure 10.11 Elemental Structure of a HTBasic CMM Inspection Program

For Test Component Number 6
10.3.5 Comparative Tolerance Analysis

The comparative tolerance analysis activity constitutes the final stage of the PEPSIM phase of the prototype PDA system and is primarily concerned with the comparison of the inspection results obtained through the execution of the inspection program on the CMM with the component nominal held within the order sub-model (see figure 10.2e). The comparative tolerance analysis is conducted transparently by a set of feature-based macros that:

i) Read the inspection results;

ii) Generate the invisible actual feature geometry;

iii) Calculate the actual feature attributes and deviations from nominal form;

iv) Assign tolerance and quality status;

v) Populate the nominal operation structure and the actual feature attribute portion of the PDA order sub-model;

vi) Produce a comparative tolerance analysis report; and

vii) Compile a PADDES fact file based on the findings of the comparative tolerance analysis.

Due to the inadequacies and limitations of the DMIS 2.1 interpreter supplied with the Ferranti Merlin 750 CMM the comparative analysis activity of the PEPSIM phase of the prototype PDA systems has been restricted to the analysis of the results produce by a CMM inspection program generated in the native HTBasic format.

The results obtained from the inspection of a component from an HTBasic inspection program are in the form of a text file, which is documented in figure 10.12. This text file contains the centre positions, size and form values for each basic geometric element that is used to construct the feature. The actual billet details are documented first followed by the basic geometric element breakdown of each inspection feature in the sequence stipulated within the CMM inspection program.

The comparative tolerance analysis is conducted by reading the feature identification number from the CMM result file and extracting the nominal feature attribute information from the appropriate feature record captured within the order sub-model. The remainder of the feature's actual data is read and the actual geometry of the feature is built-up invisibly on the graphic user interface from the actual basic geometric element information contained within the CMM results file. The corresponding actual
attributes of that feature that corresponds to the nominal feature attributes extracted from the PDA order sub-model are calculated, and compared with the feature nominals to determine the feature deviations from nominal form. These deviations are subsequently compared with the tolerance specification to ascertain the tolerance condition and the associated feature attribute quality status. The calculated actuals,
deviations, tolerance condition and quality status forms the information contained within the actual portion of the PDA order sub-model as shown in figure 10.13.

<table>
<thead>
<tr>
<th>PART_ID</th>
<th>FEAT_NAME</th>
<th>FEAT_QST</th>
<th>ACT_LGH</th>
<th>ACT_WTH</th>
<th>ACT_UFL</th>
<th>ACT_DIA</th>
<th>ACT_CBDMA</th>
<th>ACT_CBLGSH</th>
</tr>
</thead>
<tbody>
<tr>
<td>TESTS</td>
<td>FK_000</td>
<td></td>
<td>39.9274</td>
<td>39.9154</td>
<td>9.9184</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACT_CREAD</th>
<th>ACT_BREAD</th>
<th>ACT_CBDMA</th>
<th>ACT_CBLGSH</th>
<th>ACT_PNS_FX</th>
<th>ACT_PNS_FY</th>
<th>ACT_BOTX</th>
<th>ACT_BOTY</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.88835</td>
<td>50.33569</td>
<td>-0.91412</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DTH_OSTAT</th>
<th>DIA_DEV</th>
<th>DIA_T_STAT</th>
<th>DIA_QSTAT</th>
<th>CSD_DEV</th>
<th>CSD_T_STAT</th>
<th>CSD_QSTAT</th>
<th>OBL_DEV</th>
<th>OBL_T_STAT</th>
<th>OBL_QSTAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEPT</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PLMA_X</th>
<th>PLMA_DEV</th>
<th>PLY_PLA_ST</th>
<th>PLMA_QSTAT</th>
<th>PLMA_X</th>
<th>PLMA_Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4.99440</td>
<td>0.0002636</td>
<td>INTOL</td>
<td>ACCEPT</td>
<td>38.01620</td>
<td>35.98526</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PLMA_QSTAT</th>
<th>PLMA_X</th>
<th>PLMA_Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEPT</td>
<td>63.60916</td>
<td>35.04036</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PLYL_2</th>
<th>PLYL_DEV</th>
<th>PLYL_PLA_ST</th>
<th>PLYL_QSTAT</th>
<th>PLYL_X</th>
<th>PLYL_Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.02564</td>
<td>0.0009009</td>
<td>INTOL</td>
<td>ACCEPT</td>
<td>38.01620</td>
<td>35.98526</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>B_PLYL</th>
<th>B_PLYL_Y</th>
<th>B_PLYL_Z</th>
<th>B_PLYL_DEV</th>
<th>PLYL_PLA_ST</th>
<th>PLYL_QSTAT</th>
<th>CYLL_D</th>
<th>CYLL_T_ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>40.93149</td>
<td>50.03544</td>
<td>-0.91412</td>
<td>0.0009009</td>
<td>INTOL</td>
<td>ACCEPT</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CYLL_QSTAT</th>
<th>CYLL_X</th>
<th>CYLL_Y</th>
<th>CYLL_Z</th>
<th>CYLL_QSTAT</th>
<th>CYLL_T_ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEPT</td>
<td>0.0072178</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CYRL_QSTAT</th>
<th>CYRL_X</th>
<th>CYRL_Y</th>
<th>CYRL_Z</th>
<th>CYRL_QSTAT</th>
<th>CYRL_T_ST</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCEPT</td>
<td>0.0072178</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CDR_QSTAT</th>
<th>CDR_DEV</th>
<th>CDR_T_STAT</th>
<th>CDR_QSTAT</th>
<th>X_DEV</th>
<th>X_T_STAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>REJECT</td>
<td></td>
<td></td>
<td></td>
<td>-0.12132</td>
<td>OUTOL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X_QSTAT</th>
<th>X_DEV</th>
<th>X_T_STAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>REJECT</td>
<td>0.33832</td>
<td>OUTOL</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CON_QSTAT</th>
<th>CON_DEV</th>
<th>CON_T_STAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>CON_QSTAT</td>
<td>CON_DEV</td>
<td>CON_T_STAT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TRD_QSTAT</th>
<th>TRD_DEV</th>
<th>TRD_T_STAT</th>
</tr>
</thead>
</table>

Figure 10.13 Actual Feature Attribute Representation of the PDA Order Sub-model for the Closed Blind Rectangular Pocket of Test Component Number 6
Figure 10.13 depicts the actual feature attribute specification and instantiation of the actual portion of the PDA order sub-model for the closed blind rectangular pocket type feature of test component number 6. The instantiation of the actual feature attribute record of the PDA order sub-model is conducted transparently upon the completion of the analysis.

The calculated actual feature attributes and the extracted nominal feature attributes and tolerance information are collated, compiled and appended to a comparative tolerance analysis report (.ta) file shown in figure 10.14. This process is

---

<table>
<thead>
<tr>
<th>Feature Details:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type:</strong> RECT</td>
</tr>
<tr>
<td><strong>Nom:</strong> Actual Attribute Specification</td>
</tr>
<tr>
<td><strong>Length (X):</strong> 100</td>
</tr>
<tr>
<td><strong>Insertion Point (X):</strong> 50</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PK_000 Feature Details:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Feature Type:</strong> PCBR</td>
</tr>
<tr>
<td><strong>Nom:</strong> Actual Attribute Specification</td>
</tr>
<tr>
<td><strong>Size:</strong></td>
</tr>
<tr>
<td>Pocket Length: 40</td>
</tr>
<tr>
<td>Pocket Width: 40</td>
</tr>
<tr>
<td>Pocket Depth: 10</td>
</tr>
<tr>
<td>Corner Circle Diam: 10</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Location:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert Point (X): 50</td>
</tr>
<tr>
<td>X Pos Tol. Status: OUTOL</td>
</tr>
<tr>
<td>X Pos Qual. Status: REJECT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CB_000 Feature Details:</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Feature Type:</strong> HCBR</td>
</tr>
<tr>
<td><strong>Nom:</strong> Actual Attribute Specification</td>
</tr>
<tr>
<td><strong>Size:</strong></td>
</tr>
<tr>
<td>Hole Diameter: 16</td>
</tr>
<tr>
<td>C/B Diameter: 24</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Form:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hole Cylindricality: 0.05</td>
</tr>
<tr>
<td>Hole Cyl Tol. Stat. : INTOL</td>
</tr>
<tr>
<td>Hole Cyl Quality Stat. : ACCEPT</td>
</tr>
<tr>
<td>C/B Cylindricality: 0.05</td>
</tr>
<tr>
<td>Conformity (C/B - Hole): 0.05</td>
</tr>
<tr>
<td>Hole Con: Tol. Stat. : INTOL</td>
</tr>
<tr>
<td>Hole Con: Quality Stat. : ACCEPT</td>
</tr>
<tr>
<td>Base Plane Flatness: 0.1</td>
</tr>
<tr>
<td>Flat Tol. Stat. : INTOL</td>
</tr>
<tr>
<td>Flat Quality Stat. : ACCEPT</td>
</tr>
</tbody>
</table>

Figure 10.14 PEPSIM’s Comparative Tolerance Analysis Report File Extract
iterative and repeated for each feature inspected and is contained within the CMM result file. The comparative tolerance analysis is terminated once every feature within the CMM result file has been analysed.

The comparative tolerance analysis report file extract depicted in figure 10.14 illustrates the results from the analysis for the billet, closed blind rectangular pocket and counterbored features of test component number 6. The report documents each feature in turn and provides the user with a résumé of the feature nominal and actual attributes, tolerance specification, deviations, tolerance condition and attribute quality status and are extracted from both the nominal and actual portions of the PDA order sub-model.

Once the comparative tolerance analysis is complete, PEPSIM interrogates the operation plan to ascertain the operation details for each feature in the format of operation comment, operation tool/cut data and operation parameters and populates the nominal operation structure of the PDA order sub-model as depicted in figure 10.15.

![Figure 10.15 Nominal Feature Operation Instantiation of the PDA Order Sub-Model of the Closed Blind Rectangular Pocket and Counterbored Features of Test Component Number 6](image-url)
The nominal feature attributes, nominal operation attributes and actual feature attributes generated by PEPSIM provide the complete instantiation of the PDA order sub-model structure. Although there can only be one definitive instance of the component’s nominal feature and operation attributes captured by the PDA order sub-model, there is the capability of the order sub-model to capture multiple instances of the component’s actual feature attributes. The multiple instances of the component’s actual feature attributes corresponds to batch inspection of nominally identical components.

The final action of the PEPSIM phase of the prototype PDA system is the compilation of a comparative tolerance analysis fact file depicted in figure 10.16.
The comparative tolerance analysis fact file is object-oriented in nature and provides the sole source of component status information for the PADDES diagnostic phase of the prototype PDA system and the information contained within it is transparently extracted from the PDA order sub-model. This fact file specifies the PADDES class definitions of all of the construction features employed in the feature taxonomy of the PDA framework. The billet details and feature nominals, with the exception of the tolerance specification, actuals and operation list for each feature object inserted into the top face of the billet comprise the remainder of the comparative tolerance analysis file. An instantiation of a closed blind rectangular pocket object of the comparative tolerance analysis fact file is illustrated in figure 10.16, showing the complete representation of the feature’s condition. The operation tool/cut data and the associated operation parameter attributes of the nominal structure of the PDA order sub-model are decomposed into their constituent individual attributes for ease of interrogation by PADDES.

10.3.6 Manufacturing Data Analysis

The manufacturing data analysis activity is conducted by the final phase of the prototype PDA system depicted in figure 10.2f, known as PADDES. PADDES has been developed to ascertain machine dependent manufacturing error based upon the results obtain through the inspection of a component part, i.e. the comparative tolerance analysis fact file produced by PEPSIM, and the diagnostic knowledge contained within PADDES’s knowledge base. The complete diagnostic logic in the form of decision trees that has been employed to construct the rule-based knowledge representation of PADDES is documented in appendix III.

Due to the physical size of the logic encapsulated within PADDES’s knowledge base, this section will be concerned with the diagnosis of manufacturing errors based on two features of test component number 6, namely: the closed blind rectangular pocket (PK_000) and the first of the socket head screw holes (S_000) depicted in figure 10.17. The logic employed attempts to diagnose manufacturing errors produced by operations carried out on a three axis vertical machining centre and examines the size, depth, position, orientation and operation attributes of the features contained within the comparative tolerance analysis fact file. The typical feature objects extracts from the comparative tolerance analysis fact file for the closed blind rectangular and the socket
head screw hole features, as depicted in figure 10.17, are documented in figures 10.16 and figure 10.18 respectively.

Figure 10.17 Closed Blind Rectangular Pocket and Socket Head Screw Hole of Test Component Number 6

Figure 10.18 Comparative Tolerance Analysis Fact File Extract for the Socket Head Screw Hole Feature of Test Component Number 6
The three axis vertical machining centre’s manufacturing errors specified within the logic contained in the knowledge base of PADDES has been evaluated for the aforementioned closed blind rectangular pocket and socket head screw hole type features of test component number 6 through the manipulation of the feature attributes of the comparative tolerance analysis fact file. The structure of this section constitutes two major areas: (i) individual feature MDA diagnosis, and (ii) combined feature MDA diagnosis.

10.3.6.1 Individual Feature MDA Diagnosis

The individual feature manufacturing data analysis is the initial diagnosis activity performed by PADDES and involves the diagnosis of manufacturing errors in the form of tooling and component set-up error for each attribute for every individual feature documented within the comparative tolerance analysis fact file.

The feature attributes employed in the individual feature MDA diagnosis of a closed blind rectangular pocket feature within PADDES includes:

i) Combined pocket length and width;
ii) Pocket depth;
iii) Pocket position in the X axis;
iv) Pocket position in the Y axis;
v) Pocket orientation;
vii) Pocket position in the Z axis.

The logic decision trees conceived and applied in the development of the logic rules employed in the construction of the knowledge base of PADDES are documented in appendix III. The decision tree representation of the knowledge base rules for the diagnosis of tool and program based manufacturing errors for the both the length and width of a key slot closed or a closed blind rectangular pocket feature is depicted in figure 10.19.
Figure 10.19 Logic Decision Tree for Diagnosis of the Length and Width of Key Slot
Closed and Closed Blind Rectangular Pocket Type Features
The logic represented by the decision tree shown in figure 10.19 is tested by altering the feature attribute values stated within the comparative tolerance analysis fact file to invoke the desired response from the logic captured within the PADDES’s knowledge base.

The modifications to the feature-based comparative tolerance analysis fact file for a closed blind rectangular pocket (PK_000) of test component number 6 is illustrated in figure 10.16 to invoke the logic shown in figure 10.19a as follows:

- (act_pock_length 40.5)
- (pock_length_deviation 0.5)
- (pock_length_tol_status OUTOL)
- (quality_status_length REJECT)

- (act_pock_width 39.5)
- (pock_width_deviation -0.5)
- (pock_width_tol_status OUTOL)
- (quality_status_width REWORK)

These modifications specify that the length of the pocket is too large whilst the width is too small thus invoking the diagnostic response documented in figure 10.20.

As PADDES conducts the diagnostic consultation, the dialogue window and the output (.paddes) file, depicted in figure 10.20a, reflects the step by step execution of the

---

**Figure 10.20** Individual Feature Diagnostic Consultation Output (.paddes) File of PADDES for Combined Length and Width of a Closed Blind Rectangular Pocket Feature of Test Component Number 6
diagnostic process. New facts that are asserted into the PADDES working memory are employed to direct subsequent diagnosis (see figure 10.20b). It can be seen from figure 10.20 that the asserted facts for both the larger length and smaller width of the pocket initiated the combined length and width conclusion and action logic, as in figure 10.19a to be asserted.

The simulation of an oversized tool in figure 10.19b employed in the manufacture of the closed blind rectangular pocket (PK_000) can be achieved through the modification of the pocket extract of the feature-based comparative tolerance analysis fact file:

```
(act_pock_length 40.55 )
(pock_length_deviation 0.55 )
(pock_length_tol_status OUTOL)
(quality_status_length REJECT)
```

These modifications reflect that the pocket length and width deviations are equal to within certain limits and greater than 0.5 mm above nominal size. Thus the pocket length and width are both designated as out-of-tolerance and assigned a feature quality status of reject. The result of the diagnostic consultation for the oversize tool error of a closed blind rectangular pocket type feature is documented in figure 10.21.

```
Figure 10.21 PADDES Output File for an Oversize Tool Error of a Closed Pocket
```
The results of the successful consultation of PADDES for the simulated oversize tool error are shown in the dialogue window in figure 10.21a and indicates the diagnostic route whilst providing the user with a number of possible courses of remedial actions to eradicate the error. As with all the individual feature MDA diagnosis performed by PADDES additional facts, as seen in figure 10.21a, are generated during the consultation to complement the component’s object instance facts provided within the comparative tolerance analysis fact file and are employed in directing future diagnostic assertions during the consultation. The milling operation method is employed to manufacture a pocket within the prototype PDA system, however the manufacturing method employed to produce a socket head screw hole feature (S_000) is the drilling operation. The decision logic tree for the latter is illustrated in figure 10.22.

![Logic Decision Tree for Diagnosis of the Hole and Bore Diameter of a Socket Head Screw Hole Type Feature](image)

Figure 10.22 Logic Decision Tree for Diagnosis of the Hole and Bore Diameter of a Socket Head Screw Hole Type Feature
The feature attributes employed in the individual feature MDA diagnosis of the socket head screw hole feature include:

vii) Hole diameter;
viii) Bore diameter;
ix) Bore length;
x) Hole position in the X axis;
xii) Hole position in the Y axis;
xii) Hole position in the Z axis.

The modifications required within the comparative tolerance analysis fact file for the socket head screw hole feature (S_000) with an oversized bore hole diameter are shown below for the production errors of: (i) wrong tool used in the NC part program and (ii) wrong tool used on the machine. The diagnosis logic of which are shown in figures 10.22a and figure 10.22b respectively:

i) Wrong Tool Used on Machine:

(nom_sh_dia 17) (sh_dia_deviation 1.19871)
(nom_sh_depth 10.8) (sh_dia_tol_status OUTOL)
(act_sh_dia 18.1987) (quality_status_sh_dia REJECT)

ii) Wrong Tool Used in Program:

(nom_sh_dia 17) (sh_dia_deviation 1.19871)
(nom_sh_depth 10.8) (sh_dia_tol_status OUTOL)
(act_sh_dia 18.1987) (quality_status_sh_dia REJECT)

(op3_comment CBORE D18)
(op3_tool_data DRF X D18 H125 N712 V234.96 H0.0 Z-12 E2 T 'M7' 'FLAT BOTTOM DRILL M7 D18 * 125')
(op3_i_majdia 18)
(op3_i_desc FLAT BOTTOM DRILL M7 D18 * 125)

The only difference in the two scenarios is that a tooling error within the NC part program involves the use of an incorrect tool specification during the operation planning stage which is apparent within the final bore operation specification of the comparative tolerance analysis fact file as stated in (ii). The output (.paddes) file and the asserted facts generated during the PADDES consultation for both the
aforementioned wrong tool error scenarios for the socket head screw hole feature (S_000) are documented in figures 10.23 and 10.24 respectively.

Both the resultant output (.paddes) files and the asserted facts illustrated by both (a) and (b) respective windows of figures 10.23 and 10.24 indicate that PADDES has diagnosed correctly in each scenario the cause of the production error and suggest the most appropriate courses of remedial actions. It must be noted that whilst PADDES does not possess a mechanism for dealing with uncertainty within its logic representation it does possess the capability of suggesting multiple remedies within the corrective actions list.
The individual feature insertion positions in the X, Y and Z axes are analysed by PADDES to attempt to ascertain additional tooling, programming or component set-up type errors. The diagnostic logic employed within PADDES for the diagnosis of positional errors in both the X and Y axes for a socket head screw hole feature is illustrated in figure 10.25. The figure indicates the possible causes of positional errors such as when the machine is unable to machine to the appropriate accuracy, the value of which is taken from the machine tool specification, drill rigidity error, programming and component set-up errors. The asserted facts generated through the execution of PADDES's positional analysis of individual features constitutes the initial status information required for the final stage of the PADDES analysis, the combined feature MDA diagnosis.

![Logic Decision Tree for Diagnosis of the Position in the X and Y axes of a Socket Head Screw Hole Type Feature](image)

Figure 10.25 Logic Decision Tree for Diagnosis of the Position in the X and Y axes of a Socket Head Screw Hole Type Feature
The simulation modification to the comparative tolerance analysis fact file for the X axis positional anomalies of NC part programming error, figure 10.25a and the drill point walking scenario of the drill rigidity error, as in figure 10.25b, for the socket head screw hole feature (S_000) having a hole diameter of 10.5 mm of test component number 6 are documented below:

i) NC part programming error/component set-up error:

\[ \text{(act ins}_\text{pntx 15.35)} \]
\[ \text{(ins}_\text{deviation}_x 0.35) \]
\[ \text{(tol status}_x \text{OUTOL)} \]
\[ \text{(quality status}_x \text{REJECT)} \]

ii) Drill point walks:

\[ \text{(act ins}_\text{pntx 15.18)} \]
\[ \text{(ins}_\text{deviation}_x 0.18) \]
\[ \text{(tol status}_x \text{OUTOL)} \]
\[ \text{(quality status}_x \text{REJECT)} \]

The logic employed for positional diagnosis differentiates between the two error scenarios through the analysis of the positional deviations in relation to the nominal diameter of the hole. The PADDES diagnostic evaluation of the above X axis positional error scenarios are pictorially represented in figures 10.26 and 10.27.

Figure 10.26 PADDES Consultation Output of the X Axis Positional Anomaly for the NC Part Programming/Set-up Error Scenario
10.3.6.2 Combined Feature MDA Diagnosis

The main purpose of the combined feature MDA diagnosis of PADDES is to analyse the asserted positional facts generated by the individual feature MDA diagnostic activity logic. This is undertaken in an attempt to establish machine tool errors and to reinforce the diagnostic assertions produced through the individual feature analysis. The logic representation of the combined feature MDA diagnosis activity for both the X and Y axes are represented by the decision logic tree diagram depicted in figure 10.28.

The simulation modifications applied to test the positional errors in the X axis involves all of the inspected features that comprise the geometry of the component. The modifications that represent the machine tool out of calibration error, as in figure 10.28a are applied to all the primary features, i.e. features inserted on the top face of the billet only, of test component number 6 and include:

\[(\text{act_ins_pntx} \ 51.5)\]
\[(\text{tol_status_x} \ \text{OUTOL})\]
\[(\text{ins_deviation_x} \ 1.5)\]
\[(\text{quality_status_x} \ \text{REJECT})\]
These modifications to the comparative tolerance analysis fact file simulates that all the primary features of test component number 6 are out of position along the X axis by 1.5 mm. The initial asserted facts generated from the individual feature MDA diagnosis (a), PADDES consultation result fragment (b) and the additional fact asserted during the combined feature MDA diagnosis (c) for the machine out of calibration error is documented in figure 10.29.
Figure 10.29 PADDES Consultation Output of the Combined Feature Analysis of the Machine is Out of Calibration Error Scenario of the X axis

The manipulation of the attributes to simulate a component set-up error in the X and Y axis involves the modification of the X primary feature insertion positions within the comparative tolerance analysis fact file to represent a component set-up error. This is achieved through simulating a slight skew effect by giving each primary feature of the component different X and Y insertion point deviations in order to rule out the machine out of calibration in the X and Y axes error scenarios. The insertion point deviation values employed to simulate this scenario range from 1.0 mm for the left-hand features, 1.5 mm for central features and 2.0 mm for right-hand features on the
component. This assertion will be sufficient to demonstrate the component set-up logic, as shown in figure 10.28b, employed by the combined feature MDA diagnosis of PADDES. The PADDES consultation results for the combined feature MDA diagnosis as indicated in figure 10.30.

Figure 10.30 PADDES Consultation Output of the Combined Feature Analysis of the Component Set-up Error in Both the X and Y axes

For a total of 41 facts.

For a total of 42 facts.
If a machine tool error is identified as a consequence of the combined feature MDA diagnosis activity the identified error is appended to the machine dependent historical log (.hlg) file. The historical log file for the Wadkin 4/6 vertical machining centre for the aforementioned machine out of calibration error identified by the combined feature analysis is depicted in figure 10.31.

Figure 10.31 The Wadkin 4/6 Vertical Machining Centre’s Machine Dependent Historical Log
The machine dependent historical log contains a detailed specification of the machine tool, obtained from the machine tool specification contained within the manufacturing sub-model, and any machine tool dependent errors identified by the combined feature MDA diagnosis activity of PADDES. This file can be subsequently analysed to aid the production planner in both job routing and planned maintenance exercises.

The manufacturing sub-model needs only to be altered to facilitate the inclusion within the machine tool specification structure of the file names and locations of the machine dependent fault library and historical log. Similarly the PDA order sub-model need only to capture the file names and locations of the consultation output (.paddes) file for each analysed component with its actual component attribute structure.

10.4 Test Case Summary

This chapter has illustrated the applicability of the prototype PDA system to automate the manufacture, inspection, analysis of tolerances and the diagnosis of manufacturing errors for a test component, whilst operating completely transparently to the user.

Six test component configurations have been manufactured and tested but have been omitted due to brevity and the possibility of duplication.
CONCLUDING DISCUSSIONS

11.1 Introduction
This discussion brings together a number of major issues of the research reported in this thesis in order to formulate the conclusions.

11.2 The Production Data Analysis Framework
Literature surveyed in chapters 3, 4 and 5 has indicated three individual identifiable supporting research avenues that impinge upon production data analysis of manufactured components.

Product information modelling has concentrated on the integration of CAE applications through the logical accumulation of all relevant information concerning a product during the life-cycle of the product (Krause et al 1993a). The information contained within the product model is stored in a single digital form and expressed in the abstract form of functional features or manufacturing features, depending upon the model's application domain.

Automated CMM inspection research has been primarily concerned with: the tolerance representation within solid modellers, probing point and path generation, the generation of inspection code and in a number of cases the analysis of inspection results in order to determine geometric anomalies. The fragmented contributions tend to produce complicated and computationally expensive solutions that provide a major obstacle to the integration into a comprehensive system.

Structured manufacturing fault diagnosis approaches and expert systems in particular have been successfully applied for a wide variety of applications, as described in section 5.3.4. These uniquely tailored systems prove to be extremely application specific and therefore cannot be reapplied to solve problems within similar application domains.

These large individual research areas provide extremely intricate solutions which themselves provide an impenetrable barrier to the closing of the manufacturing information feedback loop. This is primarily due to their inability for integration within
The novel framework and subsequent computational environment developed by the author opens a new approach to the integration of the three literature themes and has been applied to the production data generation and analysis of feature-based prismatic components.

11.3 Machine and Inspection Planning

The conventional approach adopted by both academic institutions and software vendor companies is to treat machine operation planning and inspection planning as two distinct activities. This approach produces a plethora of individual and insular solutions that are inherently closed in nature and provide little or no integration abilities. However, the majority of the information required to conduct these activities is identical.

The novel PDA framework takes advantage of this commonality by utilising the same data contained within information models to simultaneously generate and verify machine operation and inspection plans through a single integrated system. The simultaneous activity is conducted transparently and requires little or no interaction on behalf of the user. The prototype PDA system possesses a number of limitations that include:

i) The CAD representation of the component’s geometry is captured and displayed as a wire frame model which can prove difficult to interpret for complex components;

ii) The simultaneous machine operation and inspection planning functionality are restricted to depression type manufacturing features only;

iii) The inspection planning capability is limited to one probing orientation;

iv) The probing points generated for each feature are of a fixed distribution and cannot be altered manually by the user;

v) There is no means of altering the order in which the features are sequenced within the inspection plan and subsequent CMM inspection program with the exception of employing the critical feature option;

vi) The simulation of the inspection probing path trajectory is restricted to a vector representation.
11.4 Production Code Generation

The current approach to production code generation reflects an identical image to that of the disjoint activities of machine operation and inspection planning in the sense that they are regarded as separate functions, especially within the vendor sector, as they are provided as separate software systems.

The PDA framework provides a computational environment that permits the automatic and transparent generation of both NC part and CMM inspection programs from a solitary integrated system that employs a single source of information. The approach taken for automatic generation of production code allows consistent programs to be produced without the requirement of CMM programming experience of the user.

The disadvantage of this approach lies in the fact that a single geometric and information representation is employed for both program generations. Any geometric errors induced within the CAD model will produce an equivalent error in both the NC part program and the CMM inspection program rendering the error invisible.

11.5 Comparative Tolerance Analysis

The tolerance analysis usually involves the comparison of results obtained from the inspection of a component geometry to the nominal representation contained within the CAD model description. Inspection results analysis contributions from academia have provided extremely complex environments that have taken considerable amounts of time and effort to develop (Yau and Menq 1993). However, vendor solutions have a tendency to be fragmented and rely heavily on the purchase of a number of products from differing vendors to build a complete inspection environment that produces a variety of interfacing and data duplication problems.

The novel approach adopted by the comparative tolerance phase of the PDA framework provides the vehicle for not only the geometric control of the component but is also responsible for the initiation of the process control feedback namely manufacturing data analysis. The execution of the comparative tolerance analysis activity and the documentation produced are generated automatically with no data input requirement from the user.

A major disadvantage of the comparative tolerance analysis phase is that not all of the tolerances stipulated by ASME Y14.5M are currently supported due to the scope and time constraints imposed upon the research. The supported tolerances are restricted
to a simple plus or minus representation and cover dimensional, location, and basic form tolerances.

11.6 Manufacturing Data Analysis (MDA)

There has been considerable research into the determination of machine faults by the deployment of condition monitoring techniques (El-Wardany et al 1995, Moore and Kiss 1995, Jemielniak et al 1998). The determination of manufacturing errors from inspection results has not received the same interest, which has resulted in only a few experimental contributions (Lee 1990, Anjanappa et al 1990, 1996, Rentoul et al 1994). This deficiency stems from the complexity and number of variables involved within manufacturing processes. Industrial attempts to troubleshoot manufacturing errors still rely heavily upon the use of traditional methods or SPC techniques, described in chapter 5.2 and 5.3.1 respectively, or the application of extreme dedicated expert systems described in chapter 5.3.2.

The MDA methodology involves a generic approach to the employment of a forward chaining expert system. The logic employed is restricted to the generic operations that can be processed on a CNC 3-axis vertical machining centre. This approach means that the expert system can be utilised to diagnose manufacturing errors for any CNC 3-axis vertical machining centre.

The main disadvantage of this assertion is that the confidence of the ultimate diagnosis will be reduced to that of a more specific knowledge representation. Another disadvantage is that although the input fact file and execution of the diagnostic consultation of the forward chaining expert system is completely automatic, the user does require a basic knowledge of the expert system’s user interface to conduct a consultation.

11.7 Data Resource Model Integration

The common practise within academia is to segregate product information from facility information in the form of product and manufacturing resource models and to utilise the information contained within each independently depending upon the application.

The information used to support the functionality of the PDA framework employs both order and manufacturing models. The initial information contained within both these models are employed by the prototype PDA system to generate
additional data in the form of product control and manufacturing feedback information that further enhances the model’s capabilities.
Chapter 12

CONCLUSIONS AND FURTHER WORK

12.1 Introduction
This chapter identifies the conclusions drawn from the research and test case scenario together with possible further work to extend the application of the PDA framework.

12.2 Conclusions
The conclusions formulated from this work are as follows:

i) A production data analysis framework has been proposed for the automatic generation of production code, inspection results analysis and manufacturing error diagnosis and has been shown to be of significant potential as an industrial manufacturing software tool.

ii) The effectiveness of concurrent generation of machine operation and inspection plans has been demonstrated. This innovation makes it possible to automatically and transparently generate production code from a single, consistent information resource model.

iii) The comparative tolerance analysis facility has been shown to play a major role in the transparent interpretation of inspection results and provides rapid reporting of feature-based component status information. This facility enables the simple analysis of feature-based manufactured components together with the automatic generation of manufacturing data analysis information with the minimum of interactive effort on behalf of the user.

iv) The novel application of manufacturing data analysis applied to the feature-based component manufacture provides a powerful prototype tool to diagnose manufacturing errors relating to machining, tooling, programming and set-up. This manufacturing data analysis capability is viewed by the author as having an
enhanced value to the information generated by each phase of the prototype PDA framework enabling feedback on product control and manufacturing performance thus providing information to close the manufacturing feedback loop.

v) The feasibility of developing structured supporting information models namely order and manufacturing models, have been shown to be essential for the efficient capture of component and production data for the automatic and transparent execution of the prototype PDA system.

vi) The test case reported in chapter 10 has proved the applicability of the research by testing each phase of the prototype PDA system and illustrates the significant potential of an integrated system for the simultaneous machine operation and inspection planning, production code generation, comparative tolerance analysis and manufacturing data analysis.

vii) The concept of an integrated multi-functional software tool supported by information models for the generation and analysis of component data has been developed and tested and shown to be of strong potential for use within a CAE environment. The ability to extend the feature taxonomy and associated manufacturing requirements in relation to the company needs is seen as the next major step of this research.

12.3 Further Work
Further work is required to extend the functionality of the PDA framework in the following areas:

i) Extend feature taxonomy;
ii) Apply the methodology to other component configurations;
iii) Apply in-process / post machining inspection at the machine tool;
iv) Extend the tolerance analysis to cover the complete ASME Y14.5M standard;
v) Extend the manufacturing data analysis knowledge;
vi) Apply the PDA framework methodology into a commercially available CAE system.
12.3.1 Extend Feature Taxonomy

The feature taxonomy employed by the prototype PDA systems, as described in chapter 7.2.1, comprises depression type features only. This taxonomy needs to be enhanced to include protrusion type features (Shah et al. 1988, Shah 1991) such as bosses and islands etc. An extension of the prototype system needs to contain a facility to allow for the customisation of the feature taxonomy to incorporate user defined features.

12.3.2 Apply Methodology to other Component Configurations

The prototype PDA system has been directed solely at the production data generation and analysis of 2½ D prismatic component configurations. The methodology employed within the prototype facility is equally applicable to analyse components possessing different configurations. Further research needs to be conducted to investigate the issues relating to the application of the PDA methodology for the code generation and analysis of rotational and sheet metal component type configurations.

12.3.3 Apply to In-process / Post-machining Inspection at Machine Tool

The prototype PDA system primary analysis function was the analysis of inspection results in order to achieve product control whilst providing manufacturing feedback to assist in the tighter control of manufacturing processes. The inspection results were obtained through post-process inspection of completed component parts on a CMM. The versatility of the system needs to be enhanced by applying the PDA methodology to inspect and analyse the components at the machine tool. This would involve the investigation of issues relating to applying in-process inspection techniques during the machining cycle for final cut parameter determination and post-process inspection of the component at the machine tool.

12.3.4 Extend Tolerance Analysis to Cover the Complete ASME Y14.5M Standard

Due to the scope of the research reported in this thesis the tolerances supported by the comparative tolerance analysis phase of the prototype PDA system have been restricted to the analysis of dimensional, location and basic form tolerances as documented in table 7.4. There is a requirement for the extension of the supported tolerances to cover the complete scope of the tolerances documented in ASME Y14.5M (ASME Y14.5M 1994) and in table 7.1. This enhancement would embrace the complete spectrum of
component configurations i.e. components possessing prismatic, rotational and sculptured surfaces.

12.3.5 Extension of the Manufacturing Data Analysis Knowledge Base
The knowledge contained within the fault library of the prototype PDA system contains generic knowledge for milling, drilling, boring and reaming type operations that can be executed on a 3-axis vertical machine centre. The knowledge of which has been extracted from the Society of Manufacturing Engineers' publications: Tool and Manufacturing Engineers Handbook (Drozda and Wick 1983) and Troubleshooting Manufacturing Processes (Gillespie 1988). There is a requirement to extend this knowledge to incorporate individual machine specific knowledge that can be employed to improve the accuracy of the manufacturing data analysis consultation. This machine dependent knowledge can be acquired through knowledge elicitation from the associated operative or by the addition of condition monitoring data.

12.3.6 Apply the PDA Framework Methodology into a Commercially Available CAE System
There will be a need to implement the PDA methodology within a commercially available CAE system. The author views such a commercial system as having three modes of PDA operation, namely: fully automatic for the features as implemented in this research, semi-automatic for company specific features and manually for user defined features.
REFERENCES


Bell, R.; Newman, S.T., 1996, “IT Tools to Improve the Manufacturing Performance of Metalworking SMEs”, Department of Manufacturing Engineering, Loughborough University, Project GR/L27077, Funded by the Control Design and Production Group (CDP) of the EPSRC.


CAM-I, 1999, “CAM-I Online Library”,


Coad, P.; Yourdon, E., 1990, Object-oriented Analysis, Prentice Hall Inc.


Gillespie, L.K., 1988, "Troubleshooting Manufacturing Processes", Society of Manufacturing Engineers”.


IDEF1x., 1999, “Internet Complete Reports”,
http://www.idef.com/Complete_Reports/idef1x.


Sivayoganathan, K.; Leibar, A.; Balandran, V., 1994, “Off-line Programming of Co-
ordinate Measuring Machines”, Advances in Manufacturing Technology VIII,
Proceeding of the 10th National Conference on Manufacturing Research, Edited by

Manufactured Components Utilising CAD Data”, Advances in Manufacturing
Technology IX, Proceeding of the 11th National Conference on Manufacturing
Research, Edited by Stockton, D. and Wainwright, C., Taylor & Francis, De Montfort
University, Leicester, pp678-682.

Sobh, T.M.; Owen, J.C.; Dekhil, M.; Henderson, T.C., 1994, “Industrial Inspection and
Reverse Engineering”, Proceedings of the IEEE CAD-based Vision Workshop, pp228-
235.


Spyridi, A.J.; Requicha, A.A.G., 1990, “Accessibility Analysis for the Inspection of
Mechanical Parts by Co-ordinate Measuring Machines”, Proceedings of the IEEE
International Conference on Robotics and Automation, pp1284-1289.

Measuring Machines”, Proceedings of the International Conference on Robotics and

Exchange Specification (IGES) for Digital Exchange of Product Definition Data, Part 1:
General, AS 3643.1.


http://www.sytsma.com/tqmtools/.


VDA-FS, 1984, “VDA Surface Interface Version 1.0, Verband der Automobilindustrie e.v. (VDA), D6000 Frankfurt am Main, Westendstrasse 61, FRG.


Appendix I

PEPS 3.0 Milling Expert Part Programming
System Manual
PEPS MILL EXPERT

MILL EXPERT PART

PROGRAMMING SYSTEM

Version 3.0

Camtek Limited, Camtek House
117 Church Street, Malvern, WR14 2AJ
Telephone: Malvern (01684) 892200
Fax: (01684) 892269

Publication: PEPSMEXPT
March 1995
Introduction

Short Overview

MILL EXPERT is a software package designed to generate, maintain and run a MILL EXPERT Part Programming System. An expert system comprises the customer’s product range, accepted technology and available tooling.

The main distinction of the MILL EXPERT system is its feature oriented approach. It uses a simple manufacturing language in which construction features, forming the final product, are identified and described to the computer.

A construction feature is that element of the final product which can be easily identified either from the design or the manufacturing view point. Construction features represent typical industry wide elements such as pockets, slots, holes, etc as well as particular industry and company standards.

An example of typical industry elements is a set of features covering a screw:

- a countersink hole for a screw head
- a clearance hole for a screw
- a tapped hole

A typical example of a particular company standard is a standard for bearing location in a gear box.

Construction features can be described using parameters derived from standard parts and/or defined by the product itself.

From the part definition, MILL EXPERT identifies the construction features to be manufactured, retrieves the manufacturing operations relevant to the construction features and defines them accordingly to the given parameters. MILL EXPERT decides the shape that has to be machined, how much metal has to be removed or left, which tools to invoke and what cutting feeds and speed to use. It automatically places the cutting operations in an optimum order and generates a manufacturing procedure for the part as a whole. The construction features can be put in any order, added, edited or removed. The actual machining procedure will correspond to the part definition submitted for part program generation.

All construction features are placed as figures and all figure’s instances will be identified and machined by MILL EXPERT. Figures can be placed on different job sides of the part. Job sides are identified with correspondent origins and figures coplanar with the job sides are machined.
MILL EXPERT is an optional software supplement of the PEPS CAD/CAM system of CAMTEK (UK). It is presumed that a customer has a knowledge of PEPS and is using PEPS for NC part programming.

Features and Advantages

Efficiency

The definition of features basically consists of the information contained in the drawing. It can be easily checked minimising the opportunity for human error. The same data is used for different technological stages and correspondent NC programs.

Flexibility

A part is defined as a collection of construction features. Features do not define a manufacturing sequence and may be added in any order. Any entry can be added or edited without interfering with other construction features. Enhancements in design, technology and tooling have only to be defined in the Data Base. They will automatically be invoked during the next run of any existing or new part.

Optimised and Comprehensive NC Machining

The machining procedure includes all operations relevant to the production stage from spotting or roughing to spring cuts and chamfering. All operations are properly sequenced and grouped. The program provides the most suitable technology for every construction feature and for the part as a whole, optimising performance. Each cutting tool will perform all of its applicable operations in an ordered sequence, eliminating unnecessary tool changes. The "perfect" NC program substantially increases the efficiency of NC machining and makes unmanned machining possible.

Consistent Quality

The proven machining procedures guarantee consistent quality with 'the human factor' practically eliminated. The number of necessary tools is substantially decreased.

Tool Magazine Management

MILL EXPERT provides the optimum usage of the tool magazine (tool offset storage) for available NC machines. It keeps a history of the magazine usage and uses tools from the magazine if they are available. If the tool is not available MILL EXPERT substitutes the necessary tool for the least used one.

Multisided Machining

MILL EXPERT provides machining of the part from different sides. All features coplanar with predefined job sides are processed by MILL EXPERT and are machined within an optimum machining procedure for the part as a whole.
Multipart Machining

MILL EXPERT provides machining of number of parts loaded on a machine tool table or on a pallet. All part features are processed by MILL EXPERT and are machined within an optimum machining procedure for the collection of parts as a whole.

Group Machining Optimisation

MILL EXPERT provides an optimisation of the machining of the programmed part, either one or many, either on the work table of the machine tool or on the pallet, either on one side or on different sides. Two optimisation modes are available. TOOL OPTIMISATION ON mode provides minimisation of tool changes. All operations on all parts on all sides will be performed with the tool before the tool change. TOOL OPTIMISATION OFF mode provides minimisation of the set up changes. All operations on one side are completed, tool after tool, before machining of another side.

Customization

An important feature of MILL EXPERT is its capability to expand according to the needs of the user. The basic system as supplied comprises the most commonly used construction features. As the user becomes more familiar with MILL EXPERT he will identify the typical construction features in his/her own manufacturing line and will be able to expand the library of available construction features.

User Guide

Start Job

MILL EXPERT starts a new job by definition of the billet. Billet parameters and position define placement of the coordinate system and working windows. Billet's material and hardness define cutting parameters of machining operations.

BILLET is compulsory feature and must be present. Machining of the billet is optional and all machining operations can be eliminated if required.

Programming is started by opening the file for a new or existing part definition. For a new job provide a file name of up to 6 characters long.

Positional Data and Explicit Geometry

Construction features are placed as figures. Figures can be inserted at any position and on any side of the part. A feature has to be inserted in an origin coplanar with the job side. Only features coplanar with defined job sides are processed and machined by MILL EXPERT. Positioning of the figures can be changed, they can be removed, added or edited. MILL EXPERT retrieves all figure instances and includes correspondent machining operations into NC program for all instances coplanar with defined Job Sides.
Construction features can be placed on different layers. MILL EXPERT processes and machines instances on active layers only. By switching layers On/Off you can Include/Exclude machining of the particular features and/or instances within current machining procedure.

Geometry for a construction feature can be generated automatically from its parametric definition or provided explicitly. Explicit geometry can be a group of points (G) or a curve (K). It has to be defined from the origin placed in the Insert Point of the figure.

Positional data and explicit geometry are stored in a PEPS VDM file and referenced during the design and part programming.

Geometry must be defined before it can be referenced by the user during the definition of construction features.

Define Part

Part Definition includes all construction features to be machined on Job Side(s) of the part within one manufacturing procedure. The construction features can be put in any order, added, edited or removed.

All features are identified by unique names up to 6 characters long. Default unique names are provided by MILL EXPERT depending on a feature type.

Parametric Construction Feature

A feature is defined by the set of parameters. To add a feature to the part definition use the Expert System Menu and correspondent Group Submenu. MILL EXPERT prompts for necessary data and creates a correspondent figure. Insert a feature figure selecting a correspondent insertion method.

Typical steps to add a feature to the part on a part side:

- Create or Select a correspondent origin on a side.
- Define a feature and insert it in the active side origin.
- Define Job Side using 'Miscellaneous/Define Job Side' menu option.

Note: Only construction features coplanar with predefined Job Sides are processed by MILL EXPERT and included into the machining procedure.

See User Reference for detailed information on available features.

All defined features are stored in VDM file. The list of defined features can be received using 'Features\List Features' menu option. The data stored in VDM can be printed in Print window using 'General\Print VDM Features' menu option. The first line contains the
feature definition data, the second contains a list of allocated geometry elements for the feature. The feature geometry is generated automatically based on feature's parameters.

The defined feature can be edited by selecting a desirable feature from the list and Edit button or using 'Features\Edit menu option. The edited feature is regenerated in existing positions and its manufacturing procedure is updated.

The feature can be deleted by selecting a desirable feature from the list and Delete button or using 'Features\Delete' menu option. A figure for the feature is deleted, its data is removed from VDM and its machining operations are removed from the part machining procedure. If the feature figure is deleted using PEPS Figure Delete command the feature definition still stays in VDM and feature's operations are displayed in All Feature Operations list. They can be removed using 'Features\List Features' menu option and Delete button. The feature figure can be regenerated using features\List Features' menu option and Edit button. If necessary the feature's figure can be inserted using PEPS Figure Insert command.

The machining of the features is defined by the set of Parametric Machining Operations based on feature's parameters. Operations are generated by correspondent feature M.OVM and can not be modified or deleted. To add a user defined machining operation to the Parametric Feature use 'Operations\Add Operation' menu option or 'Operation\Feature Operations menu option and Add button. If any of the parametric operations has to be modified use 'Features\Explode Feature' menu option. When feature is exploded its parametric definition is removed from VDM Parametric Features set and all its operations become user defined.

The machining of the feature can be displayed by selecting a desirable feature from the list and Display button or using 'Operations\Machine Feature' menu option. The machining is displayed in the first feature instance only. To see machining of the feature in all instances use Operations\Machine All Features' menu option.

COMMENT is an optional parameter. It is an alphanumeric string of up to 40 characters long. The string will accompany all manufacturing operations for the construction feature. It can be used to provide special information for the NC machine operator. If comment is started with 'STOP:' STOP PROGRAM command (MOO) will be issued with all operations for the construction feature.

It is possible to create and add new features to the Data Base. See Development Guide, Construction Features for instructions.

**User Defined Machining Operations**

If necessary a user defined machining operation can be added to the existing Parametric Feature or to any existing figure. To add operation use 'Operations\Add Operation' menu option. The geometry required for the operation must exist. When the operation is added to Parametric Feature the feature's geometry is displayed in cyan colour.
Each operation is completely independent and contains all data required for its execution:

- Operation Comment
- Tool/Cut Data
- Operation Data

The data of user defined operations is stored in VDM file. To list user defined operations use 'General\List VDM Operations' menu option. User defined operations are processed and displayed along with all other machining operations. All user defined operations have a unique operation number. All parametric operations have an operation number 0.

The type of the operation is selected from the list of available machining routines and a correspondent dialogue box is opened.

If the user defined operation requires geometry not provided by the feature it is recommended:

- Create or Select Origin at the insert point of the feature.
- Create necessary bounded geometry, curves and/or groups.
- Add the user defined operation(s) to the feature.

To display, edit or delete a user defined operation use 'Operations\Feature Operations' menu option and a correspondent button.

**Tool/Cut Data**

To define Tool and Cut data for the operation click 'Tool>' push button to invoke Tool/Cut data definition dialogue box. All tool types applicable to the selected operation are shown in ToolType list box. When user selects a tool type an available tool cutting materials are listed in ToolMaterial list box.

A user is required to define desirable tool diameter, tool length and other tool parameters if relevant. To see all actually available tools click 'List >>' push button. You can select the required tool or the nearest prototype and click 'OK'. The data of the selected tool is placed into Tool/Cut dialogue box.

You can type in an approximate diameter desired, length and other parameters if relevant. It is important to define a corner radius for a form cutter or an included angle for tapered tools, drills and chamfering tools. If you click 'List >>' push button MILL EXPERT will list all available tools satisfying your requirements with diameter and length within 10% of requested.

To retrieve tool and cut data from Tool/Cut Database click 'ToolLib' button. By default ExactDiameter check box is Off and Tool/Cut Database is interrogated in search for the requested or the nearest smaller tool diameter. If the exact requested diameter is required
ExactDiameter check box has to be On. If required tool is not available MILL EXPERT adds a warning to the tool description.

By clicking 'Add' push button the user can add the required tool to the data base.

**Operation Data**

The manufacturing stage for the operation has to be selected from Stages list box. When the machining procedure for the part is generated only operations for the requested manufacturing stage are included.

The machining sub stage has to be selected from SubStage list box. When the machining procedure for the part is generated the operation will be grouped with other operations and executed at the requested sub stage. The sequencing of operations depends on a tool data and an operation type. To control sequencing of the same type of operations using the same tool use Operation Priority parameter in a range 0..9. The operation priority for parametric operations is reserved as 5.

The user has to provide all data required by the operation typing it in or retrieving it from CAD data on a screen by clicking CAD help push button '>' where applicable.

**Operation Comment**

Operation Comment is compulsory and describes the operation. Click 'Comment' push button to receive a predefined operation comment and edit it if necessary.

**PEPS Macro Operation**

Use 'Operations\Add PEPS Macro' menu option to add PEPS Macro operation. The user is prompted to select a type of the operation which will be used to direct the placement of the macro within the machining procedure. The PEPS Macro dialogue box includes tool definition, manufacturing stage and the macro name for the operation. Mill Expert opens a macro body in a program window and invokes PEPS Mill Machining Menu. All REPS commands are collected in the PEPS Macro PPS_MAC. Use 'General/End PEPS Macro' menu option to end PEPS Macro. The generated macro has to be saved as name.OVM in a jobs directory and deleted from the program window. The REPS Macro is interpreted by Mill Expert as a user defined machining operation and will be processed with the right tool at the correspondent stage as a part of the complete machining procedure.

**Tool - Cut Library**

Tool-Cut library includes all the data used by Mill Expert to define a tool and operation cutting parameters according to the material type of the part, the type of operation, the tool and operation data.
Tool library includes number of inter related data sets:

- MATERIAL.DBF
- SPEEDS.DBF
- FEEDS.DBF
- TOOLTYPE.DBF
  - tool material/geometry data sets

Detailed below is a data flow involved in the tool and cutting parameters definition.

- **Material definition**

<table>
<thead>
<tr>
<th>part material</th>
<th>MATERIAL.DBF</th>
<th>material group</th>
<th>material description</th>
</tr>
</thead>
<tbody>
<tr>
<td>material condition</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Cutting parameters, set 1**

<table>
<thead>
<tr>
<th>tool type</th>
<th>SPEEDS.DBF</th>
<th>cutting material</th>
<th>surface speed m/mm preference</th>
<th>feed group</th>
</tr>
</thead>
<tbody>
<tr>
<td>material group</td>
<td></td>
<td></td>
<td>feed group</td>
<td></td>
</tr>
</tbody>
</table>

- **Cutting parameters, set 2:**

<table>
<thead>
<tr>
<th>feed group mm/tooth</th>
<th>FEEDS.DBF</th>
<th>hfeed normal</th>
<th>vfeed normal</th>
<th>hfeed good mm/tooth</th>
<th>vfeed good mm/tooth</th>
<th>hfeed fine mm/tooth</th>
<th>vfeed fine mm/tooth</th>
<th>peck</th>
</tr>
</thead>
<tbody>
<tr>
<td>tool dia mm/tooth</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tool length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Tool data**

<table>
<thead>
<tr>
<th>tool type</th>
<th>tooldata.DBF</th>
<th>teeth</th>
<th>manufacturer</th>
<th>holder VDM</th>
</tr>
</thead>
<tbody>
<tr>
<td>tool material</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dia major</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dia minor WLI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>length</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>corner radius</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>included angle</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Tool definition**

<table>
<thead>
<tr>
<th>tool type</th>
<th>TOOLTYPE.DBF</th>
<th>tool description</th>
<th></th>
</tr>
</thead>
</table>
You can change data in the tool library data sets, delete or add new entries as necessary. The updated data will be invoked by MILL EXPERT automatically on the next run of any part program.

The material data set MATERIAL.DBF is a list of materials used in the user's manufacturing environment. It establishes a link between material and cutting parameters. Type of material and material condition (hardness) are used to define a corresponding material group.

SPEEDS DBF establishes connection between material group, tool type, user preference and tool cutting material, cutting surface speed and feed group.

All tools available in a company have to be registered in tooltype data sets. For example all drills are included into DRL data set. Mill Expert is scanning tool data sets, retrieves available tools and warns if a new unregistered tool is required. If a new tool type and/or tool cutting material to be introduced a correspondent new or existing tool group has to be identified. A new tool_type group can be created using 'Add Tool Group' menu option. All new tools have to be added to the correspondent group with their parameters defined using 'Maintain Tooling' menu option. To delete unnecessary tool group delete its .DBF and .MDX files and remove correspondent record from DBASEFMT dataset using 'Maintain Data Set' option from 'General Utilities' submenu.

Depending on the requested tool type and tool cutting material MILL EXPERT interrogates corresponding tooltype.DBF data set which contains data about all available tools in a company in the group. MILL EXPERT returns result of the search in a complete tool definition. If exact tool was required and it is not found MILL EXPERT is placing a warning *** NEW TOOL' in the tool description. If the nearest smaller tool can be used and the found tool differs by more than 20% a warning ~ NEW TOOL' is added to the tool description.

For the selected tool FEEDS.DBF provides horizontal and vertical feeds in mm/tooth and a peck value depending on a feed group, major tool dia and length and required finish NORMAL, GOOD or FINE.

It is recommended that support and maintenance of the Tool-Cut Database was performed by the trained and qualified person.
Tool Magazine Management

Tool Management provides the optimum usage of the tool magazine (tool offset storage) for available NC machines. It keeps a history of the magazine usage and uses tools from the magazine if they are available. If the tool is not available MILL EXPERT substitutes the necessary tool for the least used one.

Tool management data is stored in turret layout files in the NC\machine sub directory. By default Mill Expert is using the last turret layout for the machine which is saved in TUR_LAST.DBF and is updated when a current job for the machine tool is post processed. You can save a current layout as a template turret to be used for a specific group of jobs later. If necessary an operator can select an existing layout of the previous job, of the prototype turret or an empty turret for the current job.

Turret layout file is a data set with fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TNum</td>
<td>Tool/Station Number, unique, not to be changed</td>
</tr>
<tr>
<td>ToolCode</td>
<td>Tool Type, 3 characters long</td>
</tr>
<tr>
<td>DiaMjr</td>
<td>Full Body Diameter</td>
</tr>
<tr>
<td>Length</td>
<td>Tool Length</td>
</tr>
<tr>
<td>Descr</td>
<td>Description, upto 36 characters long</td>
</tr>
<tr>
<td>RadCor</td>
<td>Corner Radius</td>
</tr>
<tr>
<td>AngInc</td>
<td>Included Angle, degrees</td>
</tr>
<tr>
<td>DiaMnr</td>
<td>Diameter WLI</td>
</tr>
<tr>
<td>Lock</td>
<td>Lock Flag, 0 - reserved, not to be used, 1 - tool in a turret (default), 2&amp;3 - for future use, 4 - station is locked. If the station is used in the current job a lock value is increased by 5.</td>
</tr>
<tr>
<td>Usage</td>
<td>Usage Counter, increased when post processed.</td>
</tr>
<tr>
<td>Order</td>
<td>Order To Fill, unique.</td>
</tr>
</tbody>
</table>

To display and maintain a turret use correspondent 'ToolCut/Turret' menu options.

Pallet Control

Standard pallet is created using 'Miscellaneous\Create Pallet' menu option. All necessary Job Sides to be defined by the user.

It is advisable to create a part figure as a whole, including into it all construction features, and load (insert) a part figure as required.

Any construction feature can be modified in an already laid-out part by using PEPS Figure Edit /End commands.

To display machining use 'Machine All Features' menu option.
**Processing Part Definition**

The defined part or part's features can be processed using 'Operations\Machine Feature' or 'Operations\Machine All Features' menu options.

If 'Machine Feature' is selected MILL EXPERT processes and displays machining for the current default manufacturing stage in one instance only. Default stage can be selected using 'General\Select Manufacturing Stage menu option.

Applicable manufacturing stages include:

- Before Stress Relieve
- Complete After Stress Relieve
- Semifinish After Stress Relieve Before Heat Treatment
- Rough Out Before Heat Treatment
- Finish After Heat Treatment
- Trim After Fitting

MILL EXPERT generates an optimum comprehensive PEPS CAM program to machine the part as a whole at the selected stage.

Stages applicable to heat treatment are activated only if billet hardness after heat treatment is provided.

The CAM program name format:

```
js, where j - job name, s - manufacturing stage code letter.
```

E.G.:

```
101 251C.TAP, job 101251, complete after stress relieve.
```

**Solid Visualisation**

**Generate Solid**

The program for solid visualisation can be generated by post processing the CAM program from MILL EXPERT. The program (.TAP) is generated in the solid post processor subdirectory (e.g. NC\PEPSSOL).

**Edit Solid**

Select PEPS Edit icon or any other text editor which does not re-format the file and open the solid program file. Edit the file as necessary providing the required resolution, sectional views, view angles, window sizes etc.
Display Solid

Select Solid Visualisation icon and open the solid program. PEPS Solid Visualisation will display metal removal as it processes the program. You can stop or finish processing and save a screen on the clipboard. You can save a picture from the clipboard as .CLP file and use it as a document.
### User reference

#### Construction Features

<table>
<thead>
<tr>
<th>Description</th>
<th>Operation Bore / Chamfer only</th>
</tr>
</thead>
<tbody>
<tr>
<td>Billet</td>
<td></td>
</tr>
<tr>
<td>Hole Bored</td>
<td></td>
</tr>
<tr>
<td>Hole Counter Bored</td>
<td></td>
</tr>
<tr>
<td>Hole Chamfered</td>
<td></td>
</tr>
<tr>
<td>Hole Drilled</td>
<td></td>
</tr>
<tr>
<td>Hole Reamed</td>
<td></td>
</tr>
<tr>
<td>Hole Socket Head Screw</td>
<td></td>
</tr>
<tr>
<td>Hole Tapped</td>
<td></td>
</tr>
<tr>
<td>Key Slot Closed</td>
<td></td>
</tr>
<tr>
<td>Key Slot Open</td>
<td></td>
</tr>
<tr>
<td>Key Slot Plunged</td>
<td></td>
</tr>
<tr>
<td>Pocket Closed Blind Any</td>
<td></td>
</tr>
<tr>
<td>Pocket Closed Blind Rectangular</td>
<td></td>
</tr>
<tr>
<td>Pocket Open Blind Any</td>
<td></td>
</tr>
<tr>
<td>Pocket Open Blind Rectangular</td>
<td></td>
</tr>
<tr>
<td>Pocket Round Blind</td>
<td></td>
</tr>
</tbody>
</table>

#### Hole Bored

![Diagram of Hole Bored]

#### PARAMETER

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature name</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Bore / Chamfer only</td>
</tr>
<tr>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>Diameter</td>
<td></td>
</tr>
<tr>
<td>Chamfer size</td>
<td>Specify 0.0 if no chamfer</td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>Machining stage</td>
<td>Before / After stress</td>
</tr>
<tr>
<td>Retract tool</td>
<td>Yes / No</td>
</tr>
</tbody>
</table>

275
### Hole Counter Bored

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature name</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Bore &amp; CBore/CBore Only</td>
</tr>
<tr>
<td>Length</td>
<td></td>
</tr>
<tr>
<td>Diameter</td>
<td></td>
</tr>
<tr>
<td>Chamfer size</td>
<td>Specify 0.0 if no chamfer</td>
</tr>
<tr>
<td>CBore depth</td>
<td></td>
</tr>
<tr>
<td>CBore dia</td>
<td></td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>Machining stage</td>
<td>Before / After stress relieve</td>
</tr>
<tr>
<td>Retract tool</td>
<td>Yes / No</td>
</tr>
</tbody>
</table>

### Hole Chamfered

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature name</td>
<td></td>
</tr>
<tr>
<td>Comment text</td>
<td></td>
</tr>
<tr>
<td>Chamfer size</td>
<td></td>
</tr>
<tr>
<td>Hole diameter</td>
<td></td>
</tr>
<tr>
<td>Retract tool</td>
<td>Yes / No</td>
</tr>
</tbody>
</table>
### Hole Drilled

**PARAMETER** | **COMMENT**
---|---
Feature name |  
Operation | Drill / Chamfer only  
Hole type | Blind / Flat / Through  
Length |  
Diameter |  
Chamfer size | Specify 0.0 if no chamfer  
Comment |  
Machining stage | Before / After stress  
Retract tool | Yes / No  
Retract tool |  

### Hole Reamed

**PARAMETER** | **COMMENT**
---|---
Feature name |  
Operation | Ream / Chamfer only  
Length |  
Diameter |  
Chamfer size | Specify 0.0 if no chamfer  
Comment |  
Machining stage | Before / After stress  
Retract tool | Yes / No  
Retract tool |  

---

277
**Hole Socket Head Screw**

PARAMETER | COMMENT
---|---
Feature name |  
Operation | Drill & CBore / CBore only  
Catalogue Id | i.e. 'M10'  
Length |  
CBore depth |  
Drill diameter |  
CBore diameter |  
Comment |  
Machining stage | Before / After stress relieve  
Retract tool | Yes / No

---

**Hole Tapped**

PARAMETER | COMMENT
---|---
Feature name |  
Hole type | Blind / Flat / Through  
Length |  
Tap diameter |  
Drill Diameter |  
Chamfer size | Specify 0.0 if no chamfer  
Comment |  
Machining stage | Before / After stress relieve  
Retract tool | Yes / No
Key Slot Closed / Open Milled

Open

Close

PARAMETER
Feature name
Depth
Length
Width
Chamfer size
Comment
Specify 0.0 if no chamfer

Key Slot Plunged

PARAMETER
Feature name
Depth
Hole diameter
Length
Width
Comment
### Pocket Closed Blind Any

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature name</td>
<td></td>
</tr>
<tr>
<td>Pocket finish</td>
<td>Fit / Semi clear / Clear</td>
</tr>
<tr>
<td>Length (bigger size)</td>
<td></td>
</tr>
<tr>
<td>Width (smaller size)</td>
<td></td>
</tr>
<tr>
<td>Z size (depth)</td>
<td></td>
</tr>
<tr>
<td>Min corner radius</td>
<td></td>
</tr>
<tr>
<td>Bottom radius</td>
<td></td>
</tr>
<tr>
<td>Chamfer size</td>
<td>Specify 0.0 if no chamfer</td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>Machining stage</td>
<td>Before / After stress</td>
</tr>
</tbody>
</table>

**Curve**

Entry XY.

**Rough out entry point.**

### Pocket Closed Blind Rectangular

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature name</td>
<td></td>
</tr>
<tr>
<td>Pocket finish</td>
<td>Fit / Semi clear / Clear</td>
</tr>
<tr>
<td>X size (horizontal)</td>
<td></td>
</tr>
<tr>
<td>Y size (vertical)</td>
<td></td>
</tr>
<tr>
<td>Z size (depth)</td>
<td></td>
</tr>
<tr>
<td>Corner radius</td>
<td></td>
</tr>
<tr>
<td>Bottom radius</td>
<td></td>
</tr>
<tr>
<td>Chamfer size</td>
<td>Specify 0.0 if no chamfer</td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>Machining stage</td>
<td>Before / After stress</td>
</tr>
</tbody>
</table>

relieve
### Pocket Open Blind Any

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature name</td>
<td></td>
</tr>
<tr>
<td>Pocket finish</td>
<td>Fit / Semi clear / Clear</td>
</tr>
<tr>
<td>X Size (horizontal)</td>
<td></td>
</tr>
<tr>
<td>Y Size (vertical)</td>
<td></td>
</tr>
<tr>
<td>Z size (depth)</td>
<td></td>
</tr>
<tr>
<td>Min. corner radius</td>
<td></td>
</tr>
<tr>
<td>Bottom radius</td>
<td></td>
</tr>
<tr>
<td>Chamfer size</td>
<td>Specify 0.0 if no chamfer</td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>Machining stage</td>
<td>Before / After stress</td>
</tr>
<tr>
<td>relieve</td>
<td></td>
</tr>
</tbody>
</table>

### Pocket Open Blind Rectangular

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature name</td>
<td></td>
</tr>
<tr>
<td>Pocket finish</td>
<td>Fit / Semi clear / Clear</td>
</tr>
<tr>
<td>X size (horizontal)</td>
<td></td>
</tr>
<tr>
<td>Y size (vertical)</td>
<td></td>
</tr>
<tr>
<td>Z size (depth)</td>
<td></td>
</tr>
<tr>
<td>Corner radius</td>
<td></td>
</tr>
<tr>
<td>Bottom radius</td>
<td></td>
</tr>
<tr>
<td>Chamfer size</td>
<td>Specify 0.0 if no chamfer</td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>Machining stage</td>
<td>Before / After stress</td>
</tr>
<tr>
<td>relieve</td>
<td></td>
</tr>
</tbody>
</table>
### Pocket Round Blind

![Diagram of Pocket Round Blind](image)

<table>
<thead>
<tr>
<th>PARAMETER</th>
<th>COMMENT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature name</td>
<td>Fit / Semi clear / Clear</td>
</tr>
<tr>
<td>Pocket finish</td>
<td></td>
</tr>
<tr>
<td>Diameter</td>
<td></td>
</tr>
<tr>
<td>Bottom radius</td>
<td></td>
</tr>
<tr>
<td>Chamfer size</td>
<td>Specify 0.0 if no chamfer</td>
</tr>
<tr>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td>Machining stage</td>
<td>Before / After stress</td>
</tr>
<tr>
<td>relieve</td>
<td></td>
</tr>
</tbody>
</table>
Feature Machining Procedure

Machining procedures are given for reference only. They can be easily customised and changed and differ from the procedures described below.

Billet

Machining of the billet depends on parametric values for correspondent operations. To eliminate the operation assign 0 value to it.

Face mill finish
If vertical size > 100 Face Mill Dia = 100
   else Dia = Vert_Size * .6

Mill around 1/8 and finish
If Depth < 50 End Mill Dia = 20
   else Dia = Depth / 2.5
If Dia > 50 Dia = 50
For a blind type mill exact depth
   else mill (depth + 5)

Chamfer 45°

Hole Bored

Drill
If Dia < 28 use Centre drill, drill (Dia - 3)
   else U-drill (Dia - 3)
Bore semifinish (Dia - 1)
Bore finish
Chamfer

Hole Counter Bored

Drill
If Bore Dia < 28, use Centre drill, drill (Dia - 3)
   else U-drill (Dia - 3)

Mill cbore
Rough cbore (Dia - 1) & mill finish
If bore_dia >= 26 use Face mill
   else use Carbide slotdrill

Semifinish bore (Dia - 1)
Finish bore
Chamfer cbore dia
Chamfer bore dia

**Hole Drilled**

Drill
If Dia < 28, use Centre drill, Drill
   else U-drill
For through hole with Drill add depth (0.3 * Dia + 1)
   or for Udrill add 5mm
Chamfer 45*

**Hole Reamed**

Centre drill
Drill with Dia = ream_dia - 0.15, Depth = ream_depth + 03*ream_dia + 1
Chamfer 45*
Ream with tool_dia = ream_dia, depth = ream_depth + 1

**Hole Socket Head Screw**

Centre drill
Drill clear dia depth = depth + 0.3*dia + 1
Counterbore with Flat bottom drill
Chamfer 45*

**Tapped Hole Entity**

Centre drill
Drill for through type depth = tapjiepth + 03*tap_dia + 1
   else for blind depth = (tapdepth + 1)/0.85
   else for flat bottom depth = tap_depth
For flat bottom drill with Flat bottom drill last 5mm
Chamfer 30*
Tap for through type depth = tap_depth + 0.3*tap_dia + 1
   else depth = tap_depth
Key Slot

Rough with Carbide slotdrill Dia = width - 0.5 (leave 0.1 on side, 0.0 bottom)
Finish with the same tool
Chamfer 45°

Key Slot Plunged

Plunge with a flat bottom drill with exact width diameter and step (Dia * .1)

Pocket Closed Blind

Rough tool
If pocket_mm_size - 2*btm_rad. <= 31 then
   slot drill with Dia = pocket_mm_size * 0.7
else Face mill with Dia:
If pocket_mm_size - 2*btm_rad > 175 Dia = 100
If pocket_mm_size - 2*btm_rad > 140 Dia = 63
If pocket_mm_size - 2*btm_rad > 60 Dia = 50
If pocket_mm_size - 2*btm_rad > 48 and pocket_max_size > 75 Dia = 38
   else Dia = 25

Intermediate tool
If Dia_Ruf >= 5 * Cor_Rad use intermediate tool
Dia_mt = Dia_Ruf / sqrt(Dia_Ruf / 2 / Cor_Rad)
If intermediate tool Dia >= 20 use Face mill else use Carbide Slot Drill

Machining of a pocket varies substantially depending on a wide range of parameters and it is advisable to separate those cases and to deal with each case on its own with a correspondent machining macro.

PCBAFAIC - Pocket, Closed, Blind, rough Asr, rough with Facemill, complete Asr, Interm tool, finish Fit

Open pocket to (depth - 0.05)
If face_mill_dia <= 50, open by Centre-dex mill with Dia = face_mill_dia +1
   else open by U-drill with Dia = face_mill_dia +1
If opened with U-drill clean bottom with Centre-dex mill with Dia
   If face_mill_dia <= 64, Dia = 25
   else Dia = 50

Rough out material with Face mill (leave 0.5/s, 0/btm)
Semifinish with intermediate tool (0.1/s, 0/btm)
With Face Mill clean corners upto btm rad (0.1/s, depth - btm_rad/btm) semifinish
   ((btm_rad + 0.1)/s, 0.0/btm)
With Carbide Slot Drill semifinish (0.1/s, 0.0/btm)
Finish contour
Tool\_dia = corner\_rad \times 2 - 1
If tool\_dia \geq 20, use Speedmax
\quad \text{else use Carbide Slot Drill}

Chamfer 45°

\textbf{PCBAFAIF} - Pocket, Closed, Blind, rough Asr, rough with Facemill, complete Asr, Intern tool, finish Fit

Open pocket to (depth - 0.1)
If face\_mill\_dia \leq 50 open by Centre-dex mill with Dia = face\_mill\_dia +1
\quad \text{else open by U-drill with Dia = face\_mill\_dia +1}
If U-drill clean bottom with Centre-dex mill
If face\_mill\_dia \leq 64, Dia=25
\quad \text{else Dia = 50.}

Rough out material with Face mill (leave 0.5/s, 0.1/btm)
Semifinish with intermediate tool (0.1/s, 0.1/btm)
If Face mill clean corners upto btm rad (0.1/s, depth - btm\_rad/btm)
\quad \text{at bottom ((btm\_rad. + 0.1/s), 0.1/btm)}
If Slot Drill semifinish around (0.1/s, 0.1/btm)

Semifinish after intermediate tool
Tool\_dia = corner\_rad*2 - 1
If tool\_dia \geq 20, use Slot Drill else use Carbide Slot Drill
If intermediate was Face mill around (0.1/s, 0.1/btm)
\quad \text{else clean corners only (0.1/s, 0.1/btm)}

Finish around
Tool diameter for contour tool\_dia = corner\_rad*2 - 1
If tool\_dia = 20, use Speedmax else use Carbide mill
Mill around (0/s, 0/btm)

Finish bottom
If tool\_dia \geq 20 use Square shoulder mill, else use Carbide mill
Finish btm (brad/s, 0/btm)

Chamfer 45°

\textbf{PCBAFAIS} - Pocket, Closed, Blind, rough Asr, rough with Facemill, complete Asr, Intern tool use, finish Semiclear

Similar to \textbf{PCBAFAIS} with an additional finish bottom cut.

\textbf{PCBAFANC} - Pocket, Closed, Blind, rough Asr, rough with Facemill, complete Asr, No intern tool, finish Clear
Similar to **PCBAFAIC** without intermediate tool.

**PCBAFANF** - Pocket, Closed, Blind, rough Asr, rough with Facemill, complete Asr, No interm tool, finish Fit.  

Similar to **PCBAFAIF** without intermediate tool.

**PCBAFANS** - Pocket, Closed, Blind, rough Asr, rough with Facemill, complete Asr, No interm tool use, finish Semiclear

Similar to **PCBAFAIS** without intermediate tool.

**PCBASANC** - Pocket, Closed, Blind, rough Asr, rough with Slot Drill, complete Asr, No interm tool use, finish Clear

Rough out material with Slot Drill (0.5/s, 0.0/btm)

Finish

\[
\text{Tool\_dia} = \text{corner\_rad} \times 2
\]

If \( \text{tool\_dia} \geq 10 \), use \( \text{tool\_dia} = \text{corner\_rad} \times 2 -1 \)

If \( \text{tool\_dia} \geq 20 \), use Speedmax

\hspace{2.5cm} \text{else use Carbide Slot Drill}

Mill around (0.0/s, 0.0/btm)

Chamfer 45°

**PCBASANF** - Pocket, Closed, Blind, rough Asr, rough with Facemill, complete Asr, No interm tool, finish Fit

Rough out material with Slot Drill (leave 0.5/s, 0.1/btm)

Semifinish around (0.1/s, 0.1/btm)

\[
\text{Tool\_dia} = \text{corner\_rad}\times 2
\]

If \( \text{tool\_dia} \geq 10 \), use \( \text{tool\_dia} = \text{corner\_rad}\times 2 -1 \)

If \( \text{tool\_dia} \geq 20 \), use Speedmax

\hspace{2.5cm} \text{else use Carbide Slot Drill}

Finish around (0/s, 0/btm)

If \( \text{tool\_dia} \geq 20 \) use Speedmax

\hspace{3cm} \text{else use Carbide mill}

Finish bottom (brad/s, 0/btm)

If \( \text{tool\_dia} \geq 20 \), use Square shoulder mill

\hspace{3cm} \text{else Carbide mill}

Chamfer 45°

**PCBASANS** - Pocket, Closed, Blind, rough Asr, rough with Slot Drill, complete Asr, No interm tool use, finish Semiclear
Similar to PCBRSANC with an additional finish bottom cut.

**Pockets Opened Blind**

Machining approach is similar to the machining of closed pockets without opening of pockets for facemill rough out.
Different cases are covered with identical POBR* machining routines.
Appendix II

Complete IDEF0 Representation of the Production Data Analysis Framework
Figure II.1 Production Data Analysis

Figure II.2 Prototype Production Data Analysis Facility
Figure II.3 Create Part Model Geometry

Figure II.4 Create Operation Plan
Figure II.5 Create Inspection Procedure

Figure II.6 Produce Production Code
Manufacturing Model

C2

CMM Spec.

Identify CMM Code Format

CMM Inspection Code Format

Post Process DMIS Format

P. Process to Native CMM Code Format

CMM Inspection Program

M1

Prototype PDA Framework (PEPSIM)

Figure II.7 Produce CMM Inspection Program

CMM Results File

Order C2 Model

C3 Geometric Algorithms

Standards C1

Expert System Specification

Tolerance Standards

Database ID

Output Results

Order Model

CTA Post File

C2

M2 Prototype PDA Framework (PEPSIM)

M1 Database

Figure II.8 Comparative Tolerance Analysis
Figure II.9 Comparative Analysis of Billet

Figure II.10 Comparative Analysis of Feature
Figure II.11 Output Details

Figure II.12 Manufacturing Data Analysis
Figure II.13 Conduct Manufacturing Data Analysis

Figure II.14 Individual Feature Manufacturing Data Analysis
Figure II.15 Combined Feature Manufacturing Data Analysis
Nob Tree for C: VDEF VVORKpDA 3. IDD
Production Data Analysis
A1. Create Part Model Geometry
   A1.1 Construct Bill of Material
   A1.2 Create Feature Geometry
   A1.3 Assign Tolerance Constants
   A1.4 Determine Printing Points
   A1.5 Add Nom. Data to Order Model
A2. Create Operation Plan
   A2.1 Identify Machine Capabilities
   A2.2 Assign Tool to Feature
   A2.3 Assign Cutting Data to Feature
   A2.4 Sequence Feature Operations
   A2.5 Simulate Operation Plan
A3. Create Inspection Procedure
   A3.1 Determine Feature Sequence
   A3.2 Review Feature Priority
   A3.3 Assign Feature Details
   A3.4 Simulate Inspection Procedure
A4. Produce Production Code
   A4.1 Identify Machine Type
   A4.2 Plot Process to Machine Dependent Format
   A4.3 Create Inspection Plan
   A4.4 Produce CMM Inspection Program
   A4.5 Identify CMM Code Format
   A4.6 Plot Process DMIS Format
A5. Comparative Tolerance Analysis
   A5.1 Comparative Analysis of Billet
      A5.1.1 Create Actual Database
      A5.1.2 Get Billet Nominals
      A5.1.3 Read Billet Insp. Results
      A5.1.4 Calculate Billet Attributes
      A5.1.5 Conduct Comparative Analysis
      A5.1.6 Complete Billet Results
   A5.2.1 Output Results
      A5.2.1.1 Populate Order Model with Feature Actuals
      A5.2.2 Define Feature Classes
      A5.2.3 Define Feature Instances
      A5.2.4 Output (PADDES) Part File
   A5.3.1 Comparative Analysis of Feature
      A5.3.1.1 Read Feature ID
      A5.3.2 Get Feature Nominals
      A5.3.3 Read Feature Insp. Results
      A5.3.4 Calculate Feature Attributes
      A5.3.5 Conduct Comparative Analysis
      A5.3.6 Analyze Tolerance Status
      A5.3.7 Complete Feature Results
A6. Manufacturing Data Analysis
   A6.1.1 Obtain Component Feature Part File
   A6.2 Obtain Machine Dependent Fault Library
   A6.3.1 Conduct Manufacturing Data Analysis
      A6.3.1.1 Individual Feature MDA
         A6.3.1.1.1 Assemble Production Error From Geometric Feature Attribute Error
         A6.3.1.1.2 Deduce Individual Feature Production Cause For Production Error
         A6.3.1.1.3 Suggest Appropriate Remedial Action For Individual Feature Pred
      A6.3.2 Output Individual Feature MDA Results
   A6.3.3 Combined Feature MDA
      A6.3.3.1 Combined Feature X-Axis Positional Error Analysis
      A6.3.3.2 Combined Feature Y-Axis Positional Error Analysis
      A6.3.3.3 Combined Feature X and Y-Axis Positional Error Analysis
      A6.3.3.4 Combined Feature Z-Axis Positional Error Analysis
   A6.4 Update Machine Historical Log
Appendix III

Decision Tree Knowledge Representation
of PADDES
Figure III.1 Feature type: DRILLED_HOLE
Feature attribute: Hole Diameter
Figure III.2 Feature type: DRILLED_FLAT_HOLE
Feature attribute: Hole Length
Figure III.3 Feature type: DRILLED HOLE
Feature attribute: Hole Position X and Y
Figure III.4 Feature type: ALL FEATURE TYPES
Feature attribute: Position Z
Figure III.5 Feature type: REAMED_HOLE
Feature attribute: Hole Diameter
Figure III.6 Feature type: REAMED_HOLE
Feature attribute: Hole Position X and Y
Figure III.7 Feature type: BORED_HOLE
Feature attribute: Hole Diameter
Figure III.8 Feature type: BORED_HOLE
Feature attribute: Hole Position X and Y
Figure III.9 Feature type: COUNTERBORED_HOLE
Feature attribute: Hole Diameter
Figure III.10 Feature type: COUNTERBORED_HOLE
Feature attribute: Bore Diameter
Figure III.11 Feature type: COUNTERBORED_HOLE
Feature attribute: Bore Length
Figure III.12 Feature type: COUNTERBORED_HOLE
Feature attribute: Bore Position X and Y
Figure III.13 Feature type: SOCKET_HEAD_HOLE
Feature attribute: Hole Diameter, Bore Diameter
Figure III.14 Feature type: SOCKET HEAD HOLE
Feature attribute: Hole Position X and Y
Figure III.15 Feature type: SOCKET_HEAD_HOLE
Feature attribute: Bore Length
Figure III.16 Feature type: KEY SLOT CLOSED
C_B_R_POCKET
Feature attribute: Length, Width
Figure III.17 Feature type:  
KEY_SLOT_OPEN  
O_B_R_POCKET  
Feature attribute: Length, Width
Figure III.18 Feature type:
- KEY_SLOT_CLOSED
- KEY_SLOT_OPEN
- KEY SLOT PLUNGED
- C_B_R_POCKET
- O_B_R_POCKET
- ROUND_BLIND_POCKET
Feature attribute: Slot /Pocket Depth
Figure III.19 Feature type:
KEY_SLOT_CLOSED
KEY_SLOT_OPEN
KEY SLOT PLUNGED
C_B_R_POCKET
O_B_R_POCKET
ROUND_BLIND_POCKET
Feature attribute: Slot /Pocket Position X & Y
Figure III.20 Feature type:
- KEY_SLOT_CLOSED
- KEY_SLOT_OPEN
- KEY_SLOT_PLUNGED
- C_B_R_POCKET
- O_B_R_POCKET
- ROUND_BLIND_POCKET

Feature attribute: Slot /Pocket Orientation
Figure III.21 Feature type: KEY_SLOT_PLUNGED
Feature attribute: Slot Length
Figure III.22 Feature type: KEY_SLOT_PLUNGED
Feature attribute: Slot Width
Figure III.23 Feature type:
KEY SLOT PLUNGED
Feature attribute: Slot Combined Length & Width
Figure III.24 Feature type: ROUND_BLIND_POCKET
Feature attribute: Pocket Diameter
Figure III.25 CONBINED MDA ANALYSIS
Feature Attribute: Single Axis X Position & Y Position

Figure III.26 CONBINED MDA ANALYSIS
Feature Attribute: Combined X Position & Y Position
Figure III.27 COMBINED MDA ANALYSIS
Feature Attribute: Single Axis Z Position
Appendix IV

List of Author’s Publications

