

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.



creative  
commons

C O M M O N S D E E D

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

 **BY:** **Attribution.** You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

**MACHINE PLANNING IN A  
PRODUCT MODEL ENVIRONMENT**

**BY**

**ROBERT IAN MARR YOUNG**

*A Doctoral Thesis*

*Submitted in partial fulfilment of the requirements*

*for the award of Doctor of Philosophy of the*

*Loughborough University of Technology*

*May 1991.*

© by R.I.M. Young, 1991.

IN  
MEMORY  
OF MY  
FATHER



## ACKNOWLEDGEMENTS

This research has been performed as part of the project "Information Support Systems for Design and Manufacture", which has been pursued at Loughborough University of Technology and Leeds University, funded by The Application of Computers in Manufacturing Engineering (ACME) Directorate and by the Department of Trade and Industry (DTI), as well as being supported by a group of industrial companies. I would like to acknowledge all those involved in the project who have provided helpful discussion in the formulation of this work, and ACME and DTI, for their financial assistance.

In particular I would like to acknowledge the help of Professor Bell, whose patient support and guidance throughout the research and the preparation of this thesis has been invaluable. I would also like to express my appreciation to Professor DePennington for his helpful advice at particularly difficult times in the preparation of this thesis. I would like to thank Trevor Downham and Richard Price for their technical advice on machining practice and also my colleagues on the research project, Mike Corrigan, Linda Lee, Gregorio Varvakis and James Gao, Paul Bell, Pete Dawson, Alison McKay and Laurence Wickens.

I would especially like to thank my wife Susan, and children James and John, for the patience and support they have shown through all the disruptions which producing this thesis has brought to their lives.



## ABSTRACT

The aim of this research was to understand and solve problems associated with the integration of a Machine Planner within a product model environment. This work was carried out in conjunction with other researchers, pursuing parallel integration issues related to pre-production proving and product data representation.

Product data representations of component level planned processes and feature level process data have been explored as sub-sets of a product data model to aid integration. Geometric queries on a cell decomposition solid model have been explored as a means of providing feature geometric interaction data, while the dimensional interactions between features have also been addressed.

Product data representations have been modelled using a prototype software tool, providing an environment for the exploration of the integration of a Machine Planner using a feature based design approach. Necessary Machine Planning functions have been implemented, using the ADA programming language, to explore the integrating capability of the product model environment, concentrating on the use of a prismatic benchmark component. Using the experimental implementation, setup and operation plans have been produced and machining part programs generated from product model representations of variants on the benchmark component. These have been successfully machined using a 3 axis vertical machining centre.

Such experiments, as well as others in conjunction with co-researchers, have shown that a product data model can provide a common base of data for the integration of a range of design and manufacturing activities.

## Table of Contents

ACKNOWLEDGEMENTS. ....	i
ABSTRACT. ....	ii
TABLE OF CONTENTS. ....	iii
TABLE OF APPENDICES. ....	viii
LIST OF FIGURES. ....	xi
1. INTRODUCTION. ....	1
2. LITERATURE SURVEY.	
2.1 Introduction .....	5
2.2 Computer Modelling in the Design to Manufacture Process .....	6
2.2.1 Current Advances Towards Automated Process Planning .....	6
2.2.2 Knowledge Representation .....	9
2.2.3 Geometry Representation .....	12
2.2.4 Features in Manufacturing Planning .....	13
2.2.5 Representing Product Data to Enable Planning .....	18
2.2.6 Concurrency in Design .....	20
2.2.7 Integration .....	22
2.3 Machine Planning Research .....	24
2.3.1 Part Program Generation for Machining .....	24
2.3.2 Planning Machining Operations .....	26
2.3.3 Setup Planning .....	29
3. INTEGRATED DESIGN AND MANUFACTURE SYSTEMS.	
3.1 Introduction. ....	33
3.2 Trends Towards Integration .....	33
3.3 The Elements of Integrated CAD/CAM Systems .....	36
3.4 Product Modelling as an Aid to Integration .....	37



**4. THE ISS PROGRAMME - AN EXPLORATION OF INTEGRATION AND AUTOMATION.**

4.1 Introduction .....	40
4.2 The Nature of the ISS Programme .....	40
4.3 An Exploration into the Integration and Automation of Interacting Activities .....	43
4.3.1 Manufacturing Code Generation .....	43
4.3.2 Manufacturing Data Analysis .....	44
4.3.3 Inspection Planning .....	47
4.3.4 Machine Planning .....	47
4.3.5 Product Data Flows in Manufacturing Code Generation .....	50
4.4 Product Data to Enable Manufacturing Code Generation .....	52
4.4.1 Workpiece Geometry and Feature Data .....	52
4.4.2 Associating Dimension and Tolerance Data with Solid Geometry. ....	54
4.4.3 Manufacturing Planning Data .....	55

**5. INTEGRATING MANUFACTURING CODE GENERATION IN A PRODUCT MODEL ENVIRONMENT**

5.1 Introduction .....	56
5.2 Capturing the Interaction Between Activities .....	56
5.2.1 The Importance of Understanding Activity Interactions .....	56
5.2.2 Problems in Defining Interactions .....	57
5.2.3 Capturing the Interactions Involved in Producing a Product .....	58
5.3 Identifying the Product Data Requirements for MCG .....	62
5.3.1 Maintaining a Focus on Integration through the use of a Common Workpiece. ....	62
5.3.2 A Summary of Data Requirements for MCG .....	63
5.4 The ISS Product Data Model .....	65
5.4.1 The Framework for a Product Data Model .....	65
5.4.2 Process Data Associated with Features .....	71
5.4.3 Planned Processes Data Associated with Components .....	71
5.5 The Use of the Product Data Model to Lay the Foundation for Integration .....	72

**6. MACHINE PLANNING - A CLOSELY COUPLED APPLICATION**

6.1 Introduction .....	73
6.2 The Role of Feature Data in a Machine Planning Application .....	74
6.2.1 Datum and Fixture Features .....	76
6.2.2 Machining Features .....	77
6.2.3 Feature Interactions .....	79
6.2.4 A Comparison of Machining Feature Recognition and Pre-definition .....	86
6.3 Machine Planning Activities and Their Interactions .....	88
6.3.1 Planning the Machining of Discrete Operations .....	89
6.3.2 Planning the Machining of Interacting Features in a Setup .....	91
6.3.3 Planning the Machining of Interacting Features across Multiple Setups .....	92
6.4 Process Data Related to Discrete Machining Features .....	97

6.4.1 General Process Data .....	99
6.4.2 The Derivation of Operation Data .....	100
6.4.3 Operation Definition Data from Machine and Cutter Combinations .....	101
6.4.4 Constraints on Operations .....	104
6.4.5 Geometric Parameters Influencing Operation Machining .....	106
6.4.6 Cutting Tool Data Related to Machining Operations .....	106
6.5 Process Data Related to Components .....	107
6.5.1 Planned Processes - Process Level Data .....	107
6.5.2 Planned Processes - Machine Level Data .....	108
6.5.3 Planned Processes - Setup and Operation Level Data .....	111
 <b>7. THE INTERACTION OF GEOMETRIC AND MACHINE PLANNING APPLICATIONS IN A PRODUCT MODEL ENVIRONMENT.</b>	
7.1 Introduction .....	115
7.1 Geometric and Machine Planner Interactions .....	116
7.3 The Methods used to Identify Feature Positional Data .....	118
7.3.1 Feature Positions on a Component .....	118
7.3.2 Feature Orientations on a Component .....	123
7.3.3 The Relative Positions of Setups .....	126
7.3.4 Updating Setup Geometry .....	126
7.4 The Method used to Identify Feature Interaction Data .....	129
7.4.1 Geometric Queries on a Cell Decomposition Model .....	129
7.4.2 Checking the Surface Around a Hole .....	131
7.4.3 Checking the Surface Under a Clamp .....	135
7.4.4 Checking for Thin Walls .....	137
7.4.5 Accuracy of Checks .....	138
 <b>8. THE DIMENSIONAL FACTOR IN A CLOSELY COUPLED MACHINE PLANNER.</b>	
8.1 Introduction .....	141
8.2 Dimensional and Machine Planner Interactions .....	142
8.3 Dimensional Representations in the Product Data Model .....	143
8.3.1 Parameter Values .....	143
8.3.2 The Relationship Graph .....	144
8.4 Inter-Feature Constraint Extraction .....	145
8.4.1 A Method of Linking the Relationship Graph to Features .....	145
8.4.2 Tolerance Analysis Requirements in Setup Planning .....	148



## 9. THE EXPERIMENTAL MACHINE PLANNING APPLICATION

9.1 Introduction .....	151
9.2 The Scope of the Machine Planner Implementation .....	152
9.2.1 Planning Functions to Explore the Integrating Environment .....	152
9.2.2 The Implemented Planning Functions .....	152
9.2.3 Fixturing Strategies and Machining Rules .....	154
9.3 The Functions to Identify Component Setups .....	158
9.3.1 Identify Features .....	158
9.3.2 Sequence Setups .....	161
9.3.2.1 Group Features by Spindle Axis Direction .....	162
9.3.2.2 Identify Geometric Relationships .....	162
9.3.2.3 Apply Setup Rules .....	163
9.3.2.4 Select Fixturing Strategy .....	164
9.3.2.5 Update Setup Data .....	165
9.3.3 Evaluate Features .....	166
9.3.3.1 Identify Cutting Tools .....	167
9.3.3.2 Calculate Processing Data .....	168
9.4 The Sequence Operation Function .....	169
9.5 The Functions Used to Generate Part Programs .....	171

## 10. THE REALISATION OF AN EXPERIMENTAL MACHINE PLANNING APPLICATION IN THE PRODUCT MODEL ENVIRONMENT - (POPULATING AND LINKING TO A PRODUCT MODEL)

10.1 Introduction .....	173
10.2 The Experimental Environment .....	174
10.4 The Population of a Product Model .....	177
10.3.1 A Framework for a Product Model .....	177
10.3.2 A Library of Machining Features .....	177
10.3.3 The Solid Application .....	178
10.3.4 Manufacturing Data .....	178
10.4 The Representation of Machining Features within the Product Model Environment .....	179
10.4.1 The Pre-definition of Machining Features .....	179
10.4.2 The Range of Machining Features Defined in the Library .....	180
10.4.3 General Parameter Representation .....	180
10.4.4 Adding Features to a Component Description .....	181
10.5 Linking the Application to the Product Model Environment .....	185
10.5.1 Interfacing to a Product Data Model .....	185
10.5.2 The Machine Planning Options in the Product Model Environment .....	187
10.5.3 The Effect of Changing a Product Data Model .....	188
10.6 The Evolution of the Product Data Model .....	189

**11. A DISCUSSION OF THE EXPERIMENTS PERFORMED IN THE EXPLORATION OF THE CLOSELY COUPLED MACHINE PLANNER.**

11.1 Introduction .....	192
11.2 The Integration of a Database, a Tool Selection Procedure and a Cutter Path Generator through the use of a PDM .....	193
11.3 The Integration of The Machine Planning Application Through the use of a PDM. ....	196
11.3.1 Building and Testing a Set of Pre-defined Machining Features .....	196
11.3.2 Sequencing Operations and Extracting NC Code for Interacting Features in a Setup .....	198
11.3.3 The Generation of Setup Planning Data from Features Interacting across Multiple Setups .....	203
11.3.4 Fixture Strategy Selection .....	205
11.4 The Integration of Manufacturing Code Generation Through the use of the PDM .....	207

**12. CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK.**

12.1 Introduction .....	210
12.2 Conclusions .....	210
12.3 Recommendations for Further Work .....	212

<b>REFERENCES</b> .....	<b>213</b>
-------------------------	------------

APPENDICES

<b>1 - Information Support Systems for Design and Manufacture (Software Utilities and Activity Modelling)</b> .....	226
1.1 Introduction .....	226
1.2 Product Data Representation Using the Structure Editor .....	226
1.2.1 Building Data Models .....	226
1.2.2 Building Models from Data Models .....	226
1.2.3 Interfacing to Software Applications .....	228
1.2.4 Parametric Representations .....	228
1.2.5 Sharing Data .....	228
1.3 Data Model Utilities used in the Research .....	228
1.3.1 General Data Structures .....	228
1.3.2 Numeric Representation and Evaluation .....	229
1.3.3 Geometric Representation and Evaluation .....	229
1.3.4 Feature Representation and Evaluation .....	230
1.4 IDEF0 - A formal Method of Exploring Activity Relationships .....	231
<b>2 - An IDEF0 Model of Producing a Product</b> .....	232
<b>3 - An IDEF0 Model of Machine Planning</b> .....	242
<b>4 - Data Representations Built in the Research</b> .....	254
4.1 Introduction .....	254
4.2 The Machine Tool Data Representation .....	254
4.3 The Cutter Data Representation .....	255
4.4 Feature Representations .....	256
4.4.1 The "pms3_8" Channel Representation .....	256
4.4.2 The Channel .....	259
4.4.3 The Closed Pocket .....	261
4.4.4 The Through Pocket .....	264
4.4.5 The Step .....	268
4.4.6 The Hole .....	269
4.4.7 The Four Corner Hole .....	270
4.4.8 The Face .....	271
4.4.9 General Parameters .....	274
4.5 Component Descriptions .....	274
4.5.1 The Bolster Plate - Variant 2 .....	274
4.5.2 The Bolster Plate - Variant 4 .....	276
4.5.3 The Glacier Reduced Sized Bearing .....	278



<b>5 - Elements of the Implemented Machine Planner Code</b> .....	281
5.1 Links to the Product Data Model .....	281
5.2 The Setups Sequence Method .....	284
5.3 The Through Clamping Strategy .....	285
5.4 The Clear Around Feature Query .....	286
5.5 The Identify Cutting Rules Procedure .....	287
5.6 The Calculation of Processing Data .....	288
5.7 The Sequence Operations Procedure .....	288
5.8 The Extraction of Machine Control Code .....	289
5.9 Machining Rules .....	291
5.9.1 Cutting Rules .....	291
5.9.2 Tooling Rules .....	292
5.9.3 Setup Rules .....	292
5.9.4 Operation Sequencing Rules .....	293
<b>6 - A User Guide to the Software</b> .....	294
6.1 Introduction .....	294
6.2 Using the Machine Planning Software .....	294
6.2.1 Files Required .....	294
6.2.2 Entering the Program .....	294
6.2.3 The Root Description of The Product Model .....	295
6.2.4 The "D" Option in the Structure Editor .....	296
6.2.5 Go to Main Menu - Option 1 .....	297
6.2.6 Tooling Database - Option 2 .....	297
6.2.7 Machine Data - Option 3 .....	297
6.2.8 User Data - Option 4 .....	297
6.2.9 Component Data - Option 5 .....	297
6.2.10 Create SDSM - Option 6 .....	298
6.2.11 Plan setups - Option 7 .....	298
6.2.12 Find Poss.Tools - Option 8 .....	298
6.2.13 Sequence Operations - Option 9 .....	299
6.2.14 Find M/c Setups - Option 10 .....	299
6.2.15 Extract NC Code - Option 11 .....	299
6.2.16 Draw SDSM - Option 12 .....	300
6.2.17 Remove drawing - Option 13 .....	300
6.2.18 Dims and Tols - Option 14 .....	300
6.2.19 Put back data - Option 15 .....	300



<b>7 - Machining Instructions and Part Programs Generated for Testpieces .....</b>	<b>301</b>
7.1 Introduction .....	301
7.2 Individual Features .....	301
7.2.1 A Closed Pocket .....	301
7.2.2 A Through Pocket .....	302
7.2.3 A Step .....	302
7.2.4 A Channel .....	303
7.2.5 A Hole .....	303
7.2.6 A Face .....	303
7.3 The Bolster Plate .....	304
7.3.1 Machining Instructions - Through Clamping Strategy .....	304
7.3.2 Machining Instructions - Down Clamping Strategy .....	306
7.3.3 Machining Instructions - Side Clamping Strategy .....	309
7.4 The Glacier Testpiece .....	312
7.4.1 Machining Instructions .....	312
7.4.2 The Part Program for Setup Number 1 .....	314
7.4.3 The Part Program for Setup Number 2 .....	316
7.4.4 The Part Program for Setup Number 3 .....	318

## LIST OF FIGURES

3.1 Mechanical Engineering Solutions .....	35
3.2 Elements of an Integrated CAD/CAM System .....	35
3.3 The Structure Editor Facilities .....	39
4.1 The Product Modelling Environment .....	42
4.2 The ISS Data Requirements and Applications .....	43
4.3 The Integration of Manufacturing Code Generation in a Product Model Environment .....	45
4.4 Manufacturing Code Generation Activities .....	46
4.5 Product Data Flows in MCG .....	51
5.1 An IDEF0 representation of Manufacture and Plan for Manufacture .....	60
5.2 The Framework of the Product Data Model .....	67
5.3 The ISS Product Data Model for a Component .....	68
5.4 Modelling Choices in a Data Structure .....	70
6.1 Examples of Feature Types in Machine Planning .....	75
6.2 Types of Fixture Feature .....	77
6.3 Types of Machining Feature .....	78
6.4 Geometric Interactions between Machining Features .....	81
6.5 Fixture and Datum Interactions with Machining Features .....	82
6.6 Dimensional Interactions between Machining Features .....	85
6.7 Accuracy Problems in Setups .....	95
6.8 The Process Data Structure for a Machining Feature .....	98
6.9 Operation Types used in the Research .....	102
6.10 Finishing Operation Terminology .....	103
6.11 Constraints on Tooling from the Operation Data Definition .....	105
6.12(a) Process and Machine Data Structures for Planned Processes .....	109
6.12(b) Setup and Operation Data Structures for Planned Processes .....	110
6.13 The Integration of Feature Identification Methods and Setup Sequencing using Planned Processes .....	113
6.14 The Integration of Machine Planning Activities using Planned Processes .....	114
7.1 Geometric and MP Interactions .....	117
7.2 Feature Positional Data .....	120
7.3 Positional Data in Fixture Datum Setting .....	122
7.4 Feature Manipulation .....	125
7.5 Setup Data Manipulation .....	127
7.6 Surface Checks using the SDSM .....	132
7.7 Checking Planar Faces in a Cell .....	137
7.8 Accuracy of Distance Checks .....	139
8.1 A 2 Dimensional Relationship Graph Example .....	145
8.2 Comparing Constraints using the SE Node Instance .....	147
8.3 Tolerance Analysis and Machine Planning Interactions .....	149
9.1 An IDEF0 representation of Machine Planning, Identify Component Setups and Sequence Setups .....	155

9.2 The Implemented Fixturing Strategies .....	157
10.1 A Product Model Framework .....	176
10.2 The Machine Planning Options in the Product Model Environment .....	186
11.1 The Integration of a Database, a Tool Selection Procedure and a Cutter Path Generator through a PDM .....	194
11.2 Dimensions of a test pocket .....	195
11.3 The Use of the Feature Operation Data Structure to Generate Machining Information .....	197
11.4 Parameter Values in Instances of the Closed Pocket Feature .....	198
11.5 The Generation of Machining Information from Interacting Features in a Setup .....	200
11.6 The Bolster Plate Workpiece .....	201
11.7 The Generation of Machining Information from Interacting Features in a Multiple Setup Workpiece .....	203
11.8 Hole Position Variants to Test Feature Interaction Queries .....	206
11.9 MCG Activity Interaction in the PDM .....	207
11.10 The Glacier Reduced Size Bearing .....	208
A1.1 An Example Object Data Model using SE Building Blocks .....	227
A1.2 An Example Instance from the Object Data Model .....	227
A1.3 Planar Half-space and Rotation Definitions .....	229
A1.4 A 2-dimensional Cell Decomposition .....	230
A1.5 IDEF0 Notation .....	231
A7 Photographs - Machining the Glacier Workpiece .....	323



## CHAPTER 1

### INTRODUCTION

In recent years the speed and quality of management and engineering decisions in manufacturing industry has been improved through the use of specific individual software programs. Each of these programs tend to be limited, however, to specific activities in their own data environment. As such there is potential for significant further improvement in industrial performance if the range of industrial activities could be integrated through a common data environment. This aim of Computer Integrated Manufacturing (CIM) has such potential benefits for industry that major worldwide research and industrial effort has been committed to its achievement. This has led to the need for new concepts in integrated systems architecture and in particular to information systems modelling to which this research has contributed.

The integration of computer aided design (CAD) and computer aided manufacturing (CAM), provides an example of the same integration challenge, but on a more localised scale. Resolving this problem would provide industry with software tools to improve the consistency and quality of engineering decisions, as well as increasing a company's capability to respond rapidly to changing market requirements. However, the problems, even at this localised level, are complex. There is a need to understand the structure of data on which decisions are based and the relationships between the many functions to be performed before an integrated design and manufacture system can maximise its potential benefits to industry.

An integrated information system for design and manufacture would benefit industry by building stronger links between design and manufacture software Applications than can be achieved through current CAD/CAM technology. Such an integrated system offers this potential by providing a common product data source for all software Applications. This in turn offers the potential to enable interactions between a range of Applications which can enhance the overall quality of design and manufacture.

The work of this thesis has addressed the problem of integrating Machine Planning within such an information system. The work has been performed as a part of the "Information Support Systems for Design and Manufacture" (ISS) research programme, which has explored the structure of product models which enable the integration of a subset of manufacturing functions concerned with the pre-production proving of part programs for machining and inspection (Manufacturing Code Generation). The programme has been pursued jointly at the Department of Manufacturing Engineering, Loughborough University of Technology and at the Department of Mechanical Engineering, Leeds University. It has been funded by The Application of Computers in Manufacturing Engineering (ACME) Directorate of the Science and Engineering Research Council and by the Department of Trade and Industry, as well as being supported by a group of industrial companies.

Within the context of the ISS programme the objectives of the author's work were:

- To contribute to the exploration and definition of product model data structures which can provide the common base of data which is essential to the integration of Manufacturing Code Generation.
- To explore and define product model data structures which can provide a common base of data to aid the integration of a Machine Planning Application.
- To explore the integration linkage from a Machine Planner to geometric and dimensional data through a product model.



The central hypothesis of the thesis is that a Machine Planner can be integrated and interact with other Applications in a product model environment. This has been explored by addressing prismatic components, described using pre-defined features, which can be machined from rectangular stock on a three axis vertical machining centre. The features considered were holes and box like shapes having a range of manufacturing methods. Machining has been considered which can be performed using parallel sided end mills, face mills and slot drills with inserted carbide cutters, as well as simple drilling operations using high speed twist drills. Standard fixture elements are considered in the form of a vice, bolts and rectangular clamps.

A key part of the initial research was to specify a necessary and sufficient Machine Planner to act as a vehicle for the research. The Machine Planner implementation was required to demonstrate experimentally that a product model could provide a comprehensive source of geometric, dimensional and planning data appropriate for Machine Planning and that such a Planner could form an integral part of an experimental design to manufacture system. It was decided to implement functions for setup, operation and NC planning to provide a base for experimentation and to use pre-defined features as an appropriate method of capturing the important characteristics of machining features.

The initial chapters of the thesis set the work into context for the reader. Chapter 2 provides a survey of relevant literature, firstly in relation to the general problems of computer modelling in design and manufacture, and secondly addressing more specific aspects of Machine Planning related research. Chapter 3 provides a view of trends towards integration, the elements required in an integrated CAD/CAM system and the importance of Product Modelling providing a basis for integration.

Chapter 4 describes the ISS programme, highlighting the areas of the author's research contribution while chapter 5 describes the research performed in conjunction with co-researchers to integrate Manufacturing Code Generation in a product model environment.

Chapter 6 describes the work performed by the author to define the nature of the lower levels of a Product Data Model (PDM) which can provide the basis for the integration of a Machine Planning Application and results in the derivation of feature and component level planning data sub-sets of the PDM. The linkage to geometric representations to extract feature positional and geometric interaction data are explained in chapter 7 while chapter 8 discusses a method of linking dimension and tolerance and feature data into a Machine Planner.

Chapter 9 explains and defines the Machine Planning functions, strategies and rules which have been implemented to explore integration within the product model environment, while chapter 10 describes the work performed to realise such an integrated system. The experimental tests performed in the course of the research are described and discussed in chapter 11, and finally research conclusions are drawn, and recommendations made for future work in chapter 12.

## CHAPTER 2

### LITERATURE SURVEY

#### 2.1 INTRODUCTION.

This chapter surveys literature of relevance to the integration of a Machine Planner in a Product Modelling Environment.

Section 2.2 introduces the reader to current research into the use of computers in design and manufacturing. Research in the areas of process planning and product modelling are described as are the use of knowledge representations, geometric representations and features. The possibilities for concurrency in design is discussed and the progress being made towards integration between design and manufacturing functions is described.

Section 2.3 provides a description of detailed aspects of machine planning research of relevance to this thesis. These are highlighted in sections covering part program generation for machining, planning machining operations and setup planning.



## 2.2 COMPUTER MODELLING IN THE DESIGN TO MANUFACTURE PROCESS.

### 2.2.1 Current Advances Towards Automated Process Planning.

The automation of process planning has been recognised for some time as being a significant part in linking computer aided design and computer aided manufacturing systems. A general introduction to Process Planning and a comprehensive review of early computer based systems is provided by Chang [27], where two principal types of process planning system are highlighted. These being variant and generative systems, where the variant approach is based on Group Technology principals while the generative approach attempts to capture process planning logic.

Variant systems provide a retrieval capability for similar process plans which can be accessed through a classification code. Examples of this approach are described by Schaffer [128], and by Link [86]. There are many other examples of this approach, with most current commercial systems being based on this principle.

The variant approach is limited in its scope for automation, as it relies on the retrieval and editing of similar plans. Research effort in recent years has therefore concentrated on aspects of generative process planning.

Generative process planning is described by Wysk [153] as "a system that synthesises process information in order to create a process plan for a new component automatically". Current experimental systems in generative process planning do not meet this rigorous definition, but have some level of in-built decision making logic. Overviews of a wide range of experimental systems can be found in several papers [27,153,146].

Because of the scale of problems to be tackled in process planning, current research work tackles only aspects of the total problem which can lead to a lack of clarity in definition of terms. However global definitions of process planning are generally consistent. Chang [27], describes process planning as "that function within a manufacturing facility that establishes which machining processes and parameters are to be used (as well as those machines capable of performing these processes) to convert (machine) a piece part from its initial form to a final form pre-determined (usually by a design engineer) from an engineering drawing". His paper also provides a more concise definition as "the act of preparing detailed work instructions to produce a part". Wysk [152], provides the definition "The subsystem responsible for the conversion of design data to work instruction" while Weill [146], states that "process planning is exclusively concerned with the selection of suitable processes and tools to transform raw materials into a finished product according to the design drawing", and provides a definition as "the process of determining the methods and sequence of machining a workpiece to produce a finished component to design specification".

It is noteworthy that process planning is a wide problem area and research work, in general, tackles only aspects of the total problem. Current work being pursued in advancing computer based process planning is generally found to be tackling the areas of research identified by Wysk [153] as:

1. The identification and capture of Process Planning logic.
2. The clear and precise definition of the part in computer compatible format.
3. The unification of captured logic and part description into a manufacturing data base.



The need for links to geometry, features and other product data, such as material specification and dimensions and tolerances is given emphasis in a range of research papers. These papers are discussed in subsequent sections of this chapter, as is the need for an appropriate means to represent planning knowledge and provide a means for integration of the various planning functions.

Due to the scale of the process planning task, research workers have generally confined their work to a highly constrained domain within the overall scope of process planning. Within the field of material removal processes, with which this thesis is concerned, research tasks are usually split to consider either rotational or prismatic parts. The problem domain is then usually further split to consider aspects of the process planning task, such as process selection, Alder [2]; tool selection, Giusti [53]; cutting data selection, Barkocy [8]; and machining sequences for holes, Matsushima [92] and Chang [26].

A recent, state of the art survey, by Alting [5] provides the view that although a great deal of research effort has been spent on automating process planning, the implementations still do not match industrial expectations. The achievement of industrially appropriate solutions is aggravated by the highly constrained domains which have been pursued. The need for an integrated framework, to enable aspects of the total process planning problem to be brought together, is given emphasis by Ham [57]. His paper suggests that production planning, process planning and operation planning are each distinctive in their goals and scope; however they are highly interactive in practice and as such should be considered as integral components, rather than separate pieces, in a manufacturing planning system. To achieve the goal of integrated planning three interrelated aspects of the subject must be addressed co-operatively. These are firstly automating existing planning functions; secondly anticipating future planning challenges; and finally suggesting a more logical planning structure.

### 2.2.2 Knowledge Representation.

As already discussed, the representation of process planning logic is fundamental to improving the levels of automation of the process planning task and is a major area of research in its own right. Research in the late 1970's and early 80's considered the use of Decision Tables and Decision Trees to represent process planning logic [4]. However, the complexity of the process planning domain led researchers to explore the use of Artificial Intelligence (A.I.), and, especially, Knowledge Based Systems (KBS) as a means to capturing process planning logic. Performing tasks which require specialist knowledge can only be performed successfully by appropriate experts. Hence software programs which perform these tasks have become known as Expert Systems. An important aspect of an Expert System is that its knowledge base should be separate, (and hence able to be updated and grow), from the program part, or inference engine, which acts on the knowledge.

Representing the knowledge in an Expert System is critical to its success. A major aspect of process planning research, in recent years, has been to explore alternative methods of knowledge representation and evaluation. The following paragraphs describe the most common representation and search techniques being explored. Detailed explanations of the technology can be found from several references [48,122,58].

The most widely used technique for knowledge representation in Expert Systems is the use of a set of Production Rules. A Production Rule takes the form of "if <condition(s)> then <action(s)>". Semantic Networks offer another representation which relates a collection of objects, called nodes, through links. Examples of common links are "is-a" and "has-a". Representing complex knowledge can be aided by the use of Frames. These are used to describe a collection of attributes that a specific object would normally possess. Attributes can have default values or pro-



cedural associations which can be used to identify the attribute value to be used. An alternative means of representing knowledge through logic is the use of Predicate Calculus. The elementary unit in Predicate Calculus is an object and statements about objects are called predicates. These predicates can either be true or false; facts can then be asserted from a range of predicates.

The ways in which a knowledge base can be searched introduce other variations in approach. The object of a search procedure is to discover a path through a problem space, from an initial state to a goal state. Forward reasoning systems proceed from a start state, while backward reasoning searches from the goal state. A range of search methods can be used, some which can be applied to problems which have compound goals. As this text is only intended to provide the briefest of introductions, the reader is referred to publications [48,122,58] for a detailed description of the techniques.

One of the first experimental systems to show real promise, with the use of KBS, was GARI [42]. The system makes use of an expert knowledge base for automating the planning of the machining sequence of mechanical parts. The knowledge is represented as Production Rules with weighted assertions, to enable conflict resolution, and experiments were carried out on a number of industrial mechanical parts of considerable complexity. The major limitations of the system were identified as the need to increase the number of rules in the system, and the use of a simple weighting of assertions to enable conflict resolution.

Since GARI a large number of papers have been published in the field of KBS application to process planning, with EXCAP [41] being one of the first U.K. based systems reported, considering production rules and probability applied to rotational part planning. Other work ranges from Kumara [81], who presents a structured framework for building Expert Systems in the process planning area, to recent

work at Heriot-Watt University, with Willis [149], which has been researching an interactive process planning assistant, with an explain facility for two and a half dimensional prismatic components. A large scale Alvey project, the Design to Product Demonstrator, has also considered knowledge representation issues in process planning, [60,61], using a truth maintenance system as a means to providing user support.

Although research is continuing in the application of KBS to process planning, the problem of computerising decision making has not yet been solved. Beigel [9], in discussing the future role of Expert Systems in manufacturing, identifies that, although they still hold promise for the future, there are many problems to be tackled, and current systems are narrow in scope. Furth [49], also believes that KBS holds promise for the future, but identifies major areas for research in process planning as CAD/CAM interface, knowledge base build-up, and interfaces to production control systems.

A specific Expert System for Operation Planning on machining centres, ESOP, [44], identifies that although a "stand-alone" Expert System provides some benefits there is a serious need for integration with design, and that a more comprehensive systems approach is necessary.

Adiga [1] uses a review of current approaches and considers his own experiences in knowledge representation and feature recognition to identify the challenges for research, before practical implementations materialise. He concludes that researchers must look beyond simplistic rule based systems. Issues which should be addressed are those of knowledge acquisition, knowledge representation, geometric modelling and feature recognition, as well as considering a global view of planning, control and integration, programming tools, learning and execution efficiency.



### 2.2.3 Geometry Representation.

The advent of computers, plotters and appropriate graphical display devices led the way to a wave of computerisation of drafting in the 1970's. Although these systems have been shown to be valuable industrial tools they are limited in their ability to automate mathematically well defined tasks, Requicha [120]. The development of solid modelling systems overcame that problem, in principle, by providing an informationally complete representation of solids, such that well defined geometric properties can be calculated automatically, [120, 121].

The main representation schemes used in current systems are boundary representations (B-Reps) and Constructive Solid Geometry (CSG). B-Reps are represented as unions of faces, with each face represented in terms of its boundary of edges. CSG models are built by the union, intersection and difference of simple, or primitive, solids such as blocks, cylinders and spheres. A brief comparison of these representations can be found in several references [120, 121, 55], and a more detailed description of solid modelling in other publications [90, 102].

Cell decomposition is a further type of representation which can be used in conjunction with CSG or B-Rep models. In a Cell decomposition model a solid is represented as a union of quasi-disjoint cells, where each cell may have a distinctive shape, providing it is homeomorphic to a sphere [143]. Triangulations are the simplest form of cell decomposition. Spatial Enumeration is a cell decomposition based on box like shapes where the cells may be of uniform size or of varying size. A representation of this type has been used in the research presented in this thesis, called a Spatially Divided Solid Model, Oliver [105].

Solid modelling has the potential to support the automation of many technical tasks in industry, from detailed strength analysis to planning of machining and assembly, Walker [144]. However, although solid modellers can provide geometric data in the automation of manufacturing tasks, other data must also be taken into account. This includes part material and tolerances imposed on dimensions Pratt [114]. Further there are technological considerations linked with shapes (features) which provide implications for manufacture. Understanding feature technology has become recognised, by Brimson [20], as a significant step in providing the link from model descriptions to planning applications. A wide range of research work has been performed in this area and is discussed in the following section. Solid modelling, although an important aspect in achieving automation of manufacturing tasks has been recognised as only a part of the representation of product data needed in design and manufacture.

#### **2.2.4 Features in Manufacturing Planning.**

Brimson [20] suggests feature technology as the link between CAD systems and manufacturing processes. Part features are described as "standard descriptions of a parts geometric characteristics, which should be independent of company or function and readily transformed into manufacturing operations and sequences of manufacturing operations". In his paper the need to define the format and structure of data to be shared between geometric modelling and other components of a CAM system. He explains the development of feature technology as providing an intermediate level of detail between a Group Technology code and the full part geometry, which is more suited to the needs of manufacturing. As well as describing features, the relationship of features to raw material must be defined, as must a hierarchy of features, to obtain relationships between them.



Feature related research falls into two principal categories: feature recognition, which aims to assess a geometric model to identify features; and feature based design systems, where the user is constrained to build a model using pre-defined features. In addition to these two principal categories, interactive feature definition have also been used in planning research, which allows a user to specify features on a model.

XCUT, described by Hummel [64], is a planning system which will generate process plans for the production of machine piece-parts, given a feature-based part description. These features were initially identified interactively from a solid model interface, where solid modelling was identified to be useful in identifying wall thickness constraints, tool clearance problems and path planning. In a later paper, Hummel [65], features are described as objects in an object oriented programming language, where "message passing" is described as an important aspect of feature representation in providing flexibility of definition. This flexibility is emphasised as it is thought "improbable that a universal set of feature definitions, which will satisfy the needs of all manufacturers, can ever be developed".

Mantyla [89] proposes the use of pre-defined features by the designer, and discusses the problems of capturing manufacturing information without constraining the designer, suggesting product modelling as the route to achieve this. In his paper, a feature is defined as "a collection of geometric entities that has some significance as a whole". The work considers the representation of features as objects and experiments with frames and production rules as the means of generating process planning information. The object-oriented approach is again found to be powerful and flexible, but the author identifies a need to capture technological information in product models.

Mill [101], describes research which does not use CAD based features, but aims to produce a form feature description language and experiment with semantic networks and production rules as a means to decision making in process planning.

The topic of automatic recognition of features is a subject of significant research, where many researchers have made contributions. Work goes back to the mid 1970's, with Grayer [54] describing an approach to building up laminae to be removed from a billet to achieve the component described. In the same period Woo [150] described an approach to identifying cavities on a model and matching these to appropriate machining shapes. Since then a large amount of research effort has been expended on the problem. Li [83] provides a review of a range of this work, highlighting the significance of this area of research to CAD/CAM integration, and the need for much more work. The following paragraphs describe some of the recent work in this field.

Walker [144] describes a feature based methodology which recognises holes from a wireframe model and automatically generates a process plan. VanHouten [140] describe a feature based computer aided process planning system, PART, which automatically recognises manufacturing features. This has been achieved by specifying patterns for features in a database and then analysing a boundary representation model to recognise these patterns, their hierarchy and the parameter values which relate to each. This provides the input to the process planning module of PART.

Henderson [59] describes a system capable of recognising swept volume holes, slots and pockets, and identifies future work in the areas of recognising a more diverse range of features, and resolving conflicts between model definitions which provide ambiguous results.



Wang [145] describes an algorithmic approach to identifying machined surfaces of symmetric rotational parts. Choi [30,31] describes an algorithm for recognising machined surfaces from a 3D representation of a workpiece by mapping from "Elementary Machined Surfaces" (such as holes, slots, and pockets) to simple process cycles for each. Problems with this approach arise when there are patterns which cannot be recognised as Elementary Machined Surfaces. Problems have also been identified with global constraints such as clamping requirements and interference between patterns.

Jared [72] describes research being pursued to investigate a feature recognition system, based on feature grammars, and separate recognition code, where feature grammars consist of production rules operating on a face-edge-vertex graph of a boundary representation modeller.

Joshi [73] discusses a project aiming to integrate design and process planning using A.I. techniques and discusses a CAD interface with respect to automated feature recognition. The CAD interface requirements are described as:

1. To determine the raw material to be removed.
2. To identify the machined faces of the part.
3. To recognise features formed by the machined faces.
4. To determine the tool approach directions for machining.
5. To obtain precedence between features based on geometry.
6. To decompose the raw material to be removed into smaller sub-volumes

The feature recogniser uses a forward chaining scheme to reduce the computational effort involved, and features are described as relationships between concavity and convexity. Analysis of the resulting attribute adjacency graph leads to feature identification, currently of holes, slots, steps, blind slots, blind steps, and pockets.

The information, produced by the recogniser, is then used in the generation of process plans containing processes, process sequence and setup information.

As can be seen, above, there are many approaches being taken to resolving the feature recognition problem. Even the most appropriate modelling system to use is in question, with Woodwark [151] favouring CSG modelling and Pratt [115] arguing in favour of Boundary Representation models. Although in Pratt's paper he accepts that CSG modelling has some value in the design by features approach. Further Pratt identifies the need for further research in feature interaction, the representation of dimensions and tolerances, and automating manufacturing planning linked to NC code generation.

Chung [33], identifies feature interaction as an area requiring further research when taking a design by feature approach applied to the investment casting process. Karinthi [77] consider overlapping features and applies an "algebra of features" to a model which produces a range of alternative feature sets from a geometric description.

The product modelling tool described by Bloor [16] has a capability to pre-define features. This has been used in the Machine Planning research of the author, to explore the representation of manufacturing data which should be associated with features to enable the integration of a Machine Planner. Feature interaction has also been considered in the work of this thesis, and from this, a paper has been published [156] which uses geometric queries on a cell decomposition model to identify feature interaction problems in fixturing.

### 2.2.5 Representing Product Data to Enable Planning.

Researchers have, for some years now, recognised the benefits of three dimensional solid modelling in aiming to provide a complete geometric description of a component, Mullineux [102], especially as a base for providing information for other activities such as manufacturing planning. Solid models offer only the geometric part of the data required to describe a product, however, and do not include other data generally required by a planner, such as dimensions and tolerances, material specification and feature data.

Recent research by Spur [133], and by Iwata [69], has identified the need to use an information system which can support the data processing of information required in CIM. Such an information system should allow models of products to be built which not only handle geometric data but can also handle wider ranging information concerning a product at each stage in its life, from design through planning to manufacture and beyond.

Kimura [78], describes an early database system for handling Product Models, highlighting structure and information flow in mechanical design and manufacture. The process planning aspect of this product modelling database relies on a retrieval approach to aid the user.

Lillehagen [85], describes a Product Model as "the entire body of information necessary to computerise the design and production of an object", and concludes that the currently available methodologies for product modelling have serious shortcomings.



Sata [127] proposes an integrated approach to process planning with the product design process. His paper suggests a product model as a model to describe design objects and manufacturing environments; that product models are generated by the design process and used by the manufacturing preparation processes. This however does not allow for the enhancement of product model information by manufacturing preparation processes. The work he reports uses a solid modelling technique to represent product geometry and form features to realise machining operations. These are then manipulated by a "plan generator" based on sets of Production Rules. They conclude that form features are important properties of objects; Production Rules are limited in representing all the required planning methods; and there is a need for a hierarchy in process planning.

The representation of dimensions and tolerances, as they relate to a product model, is also essential information to enable successful planning to be performed. Linking dimensions and tolerances to two dimensional representations is readily available in today's commercial CAD draughting packages. The problem of relating dimensions and tolerances to three dimensional models is more complex, however, and no commercial systems currently exist. A range of alternative approaches to linking dimensional data to solid models are being researched as described in the following paragraph.

Imamura [67] illustrates, for simple two and a half dimensional components, how dimensions can be linked to geometry, such that as dimensions change the geometry is updated accordingly. Ranyak [117] uses features and datum reference planes to dimension and tolerance a boundary representation model, while Wickens [148] has experimented with a Relationship Graph related to the faces of a Constructive Solid Geometry model. Shah [129] describes a structure for integrating a tolerance modeller which can be combined with a geometric modeller and a feature modeller, using evaluated boundary entities. Kimura [79] describes research using predi-

cate logic to represent dimensional constraints. He highlights the need to make user access and control of such a system more straight forward, as well as the need to provide a flexible geometric reasoning system, which can manipulate the dimensional constraints.

Bloor [16] describes an experimental "Product Description System" to create product model data structures, handle data and allow access to application software and the user. The data structures under consideration include representations for geometry, features, material specification and dimensions and tolerances. The work presented in this thesis has contributed to this system by exploring product data structures which enable the integration of Machine Planning functions in a product modelling environment.

Mantyla [91], in discussing directions for future research in product modelling, identifies one of the key problems as the requirement for a thorough understanding of design and manufacturing processes. Spur [135], extends the product modelling concept to include a resource model and a control model. He then considers modelling the entire information within a manufacturing enterprise as a basis for CIM integration.

### 2.2.6 Concurrency in Design

The concept of concurrency in design is to provide simultaneous input, from a range of disciplines, to resolve problems in product design. Simultaneous Engineering is an alternative term sometimes used to describe the same concept. The object of such an approach is to develop better products by having various experts working together on a problem, rather than the traditional sequential approach of passing the problem from design engineers to planning engineers etc. Vasilash [142] offers a useful introduction to the concept, providing a comparison with traditional



methods.

The ability to aid design teams to achieve concurrency, through the use of computer software, is beginning to be explored in the research community. Cutkosky [37] takes the approach that products should be designed in terms of manufacturing plans. He describes a system, FIRST-CUT, which is proposed to operate in different design modes. Each mode allows alternative experts to act on a process plan representation. Further, he argues [38] that a combination of features and a process representation is the correct foundation upon which to build a design tool. Alder [2] describes a similar designer aid for simultaneous engineering, called DAISE, which considers the application of knowledge bases to evaluate a design. These knowledge bases cover the domains of functional design, design for assembly, and design for injection moulding. Yoshimura [154] proposes an integrated design process providing simultaneous decisions for both product and process design. His paper considers a systematic design optimisation method applied to robot design. Nevins [104] states that successful manufacturing requires integration of many previously compartmentalised activities, plus new knowledge and greater understanding of manufacturing processes, product design methods and manufacturing system design methods.

Lu [87] describes the need to identify a clear model describing the decision making activities in simultaneous product and process design. Further he identifies the need for new computer based tools to support such cooperative decision making activities. Spur [134] also identifies the need for new software structures to enable interaction between computer aided design, planning, production and quality control. His paper discusses the possibilities of merging design and manufacturing knowledge through the use of features, product models, and knowledge based application modules.



### 2.2.7 Integration.

Achievement of concurrency in design requires that integration between the activities involved, such as functional design, planning for manufacture, planning for assembly, etc, can be achieved. Although there is greater understanding of the problems involved in the route to successful integration between a range of activities in CIM, the solutions have not yet been proven. The advances in product modelling and their links to manufacturing planning are, however, beginning to show the way forward.

ESPRIT project 2165 entitled "The Integrated Modelling of Products and Processes using Advanced Computer Techniques" (IMPPACT) [68] is addressing the logical integration of CIM modules by considering a complex system as a formal model in terms of functions and information. It is exploring work in design and feature modelling, linked to process modelling and operation planning, concentrating on the design and manufacture of sheet metal parts and ships propellers.

Tattersall [138] argues for a data oriented approach which involves the construction of a strategic data model showing key activities and data relationships. He argues that planning for integration requires the development of a data model and the identification of major data flows and application areas. Smith [132], in modelling CIM systems, highlights the need to understand the activities being performed and their interrelationships, as well as identifying the flows of information between them. He proposes that this understanding then provides the basis from which to create a data model. Further he identifies IDEF0 as a valuable activity modelling tool which can aid the generation of this understanding.

IDEF0 stands for ICAM Definition language, a representation technique based on a Structured Analysis Design Technique (SADT) [123,124]. The name follows from the US airforce ICAM programme, where the technique was used to consider an overall concept for CIM [105]. The technique has been used in industry to aid in the understanding of the relationships between different activities in a company's business [80]. Colquhoun [34] has explored the use of IDEF0 to link design and manufacture in a CIM environment and identified that IDEF0 is a powerful tool to aid the rationalisation of complex situations and that it provides a structured method of identifying the role and interfaces of computerised manufacturing activities. The technique has also been recently used by Ferreira [47], on behalf of CAM-I, to explore a conceptual model for process planning.

The importance of data modelling in integration is given further emphasis by the work of the ISO on standards for product data. Two organisations, the Standard for the Exchange of Product Model Data (STEP), and Product Data Exchange Standard (PDES), are now working jointly towards the aim of producing a standard for product data. Their work should result in a standard which can be used in the integration of software which uses product model data, such that data can be exchanged between separate software packages by keeping in line with the standard. This supercedes work on Initial Graphics Exchange Standard (IGES) which was concerned with the exchange of geometric data only. Part of the PDES/STEP work [108], is concerned with an information model for form features. It is addressing the shape representation information rather than the non shape data which might be attached to the feature. Another aspect of the work is concerned with the representation of process planning data within a product model [109].



The ISS research programme , and the research reported in this thesis, contribute to integration research. This contribution takes the form of a greater understanding of product modelling as an enabling mechanism for the integration of Manufacturing Code Generation activities.

### **2.3 MACHINE PLANNING RESEARCH.**

#### **2.3.1 Part Program Generation for Machining.**

Numerical control (NC) part programming is concerned with producing instructions which can be interpreted by a machine controller to control the motion of a machine tool. A useful introduction to this, including its history and current capability, is provided by Groover [55] and also by Rapello [118]. Following the introduction of NC machine tools, the first significant aid to part programmers was APT, an English like language used to describe the required tool motions. There are now a number of commercial variations on this such as Exapt, Compact 2 and GNC.

Since the introduction of APT, the major step towards automation has been the use of interactive graphics providing an integration link between two dimensional Computer-Aided Design (CAD) and NC part programming. When a CAD description of a part exists, interactive graphics can be used, by a part programmer, to identify the specific shapes or profiles in which he is interested. These can then be used to generate geometry statements in a part programming language. There are many such commercial systems available today, of which Applicon Bravo3 [6], Dogs NC [106] and Peps [22] are examples. Commercial NC code generation systems are beginning to appear which are based on three dimensional solid geometry



representations. STRATA [137] is such a system and adds collision avoidance checks and a simple user interface to aid NC code generation. It also introduces the possibility of the use of feature description to improve levels of automation.

The levels of automation in current commercial systems is low, with many decisions being required of the user. These include: the identification of the stock as it relates to the component to be machined; the setups required; the selection of cutters and their cutting parameters; and the order in which machining elements are to be performed in a setup. Considering the decisions to be made in part program generation it can be seen that there are a number of similarities with process planning. Groover [55], identifies the difference as being that process planning is concerned with the preparation of a routing sheet, while part programming plans the process for the portions of a job to be done on an NC machine. A routing sheet lists the sequence of operations to be performed on the workpiece, listing the machines through which it must be routed. A part programmer should therefore be considered to be both involved with the generation of NC code, and also the production of machining instructions for a specific machine tool.

The majority of current research which considers decision making in manufacturing planning concentrates on process planning, viewing part program generation as a subsidiary requirement. However, much of the research is simplified in scope to consider machine specific problems. This introduces greater similarity with part program generation than process planning, using the definition described in the last paragraph.

Most recent research directly involved with NC code generation has been concerned with the use of geometric model information, rather than the representation of technological information. Examples of this work are Sata [125] who makes use of a solid model to perform tool collision checking as well as generate a cutter

path, through the use of interactive graphics; Yeun [157] describes an cell decomposition method which automatically generates a rough machining cutter path for a workpiece, using one cutter and from one cutting direction. Armstrong [7] also describes a method for automatically generating paths from solid model descriptions using only one cutter for roughing, but cutting directions are also sequenced on the basis of material removal capability from each direction.

The automatic generation of NC part programs from a feature based component description is proposed by Magill [88]. This work aims to use a set of parametric procedures to select the optimum cutter, cutting speed and feed rate and generate tool paths for each feature. Most research related to planning logic has been centred in process planning. However, Laing [84] describes work which uses an Artificial Intelligence (A.I.) approach, considering the interaction of a number of machining contours. Priess [116] also considers A.I. techniques to generate part programs for discrete shapes on a component.

Ssemakula [136] in addressing the problem of automatic generation of NC part programs, bypasses the planning problem, by assuming that this has already been achieved through a process planning system. They address the use of process plan data, in conjunction with a part programming system, to automatically generate part programs. A similar approach is also reported by Genord [51].

### **2.3.2 Planning Machining Operations.**

Operation planning has a narrower focus than process planning, being concerned with planning the work to be done on a component at a particular workstation, as opposed to planning the part as a whole which may require routing through a number of machines [55]. In some work an operation is considered to be the machining to be done on a particular machine; in others an operation is considered



to be a piece of machining to be done by one cutter. In this research the definition has been used that an operation is a discrete piece of machining which is to be performed by one cutter. Operation planning is then concerned with how the operations to be performed on a machine should be planned.

Carlier [24] describes a system, MOPS, which has had a significant formative effect on the research reported here. The paper describes an early system relating to operation planning on machining centres, which considers the break down of the machining into operations, the selection of tools and cutting conditions, and the interactions between tool selection and operation sequence. More recently Detolenaere [44], from the same laboratory, reported on a follow up system, ESOP, which considers the application of Expert Systems technology to operations planning. Their conclusions highlight the need for geometric information during planning.

The need for geometric information has been considered by Joshi [73], who describes a system which uses "feature precedence" information when deciding on how to produce particular setups. Also Willis [149] describes research aimed at integrating geometric information into an Expert System approach to planning, using machining strategies and setup strategies. The reported in that paper concentrates on pocketing, or slot drilling of components and addresses geometric tolerances on machining, identifying feature interactions as a key problem area. The Quick Turnaround Cell (QTC) work of Purdue University [28], which has some similarity with the ISS project, considers many aspects of planning and NC code generation. These include the use of vice fixtures and the automatic generation of NC code. In the QTC work the generation of cutter path information is seen as a subsequent task to planning, rather than an integral part of the planning problem.



Considerable research has been performed in the area of cutter selection and cutting data determination. The paper of most significance to the author's research, relating to tool selection, is the work by Carlier [24], which describes a methodology which has been adopted by the author. This is to find a set of alternative tools which may be used for a machining operation and then to select the actual tool to be used at the stage of sequencing the operations. This allows the effect of neighbouring operations to be included in the tool selection decision. A further paper by Melkote [95] describes the effect of tip geometry on selecting tools for facing and end milling. He illustrates how a range of suitable tools can be narrowed down by considering such constraints. Chen [30] also considers process constraints on cutting conditions to identify minimum costs, which are then used as a basis for tool selection.

Selecting cutting data can be achieved through the use of look up tables or through the use of equations relating to the required life of the tool. Metcut's machinability data handbook [99] provides an example of the look up approach, with [98] illustrating a computer based version of the same approach. The use of extended versions of F.W.Taylor's tool life equation has also received considerable attention. PERA have expended considerable effort in this area, describing their Macbank machinability data approach in several reports [100,102,111,112].

Tool life equations can be used to provide flexibility in the data values generated. This is illustrated in the work of Hinduja [62] who identifies cutting data for turning operations based on different tool life criteria. These criteria being maximum removal rate or minimum cost production. A theoretical relationship between feedrate and surface finish, described by Boothroyd [19], can also be used to provide a link from surface requirements to machinability data.

A summary of industrial requirements for machinability data by Kahles [74] highlights a wide range of views of machinability data requirements for advanced machining systems. He reports that the need for reliable data on tool life, chip control, dimensional accuracy, surface finish and surface integrity is generally accepted. However the problem of achieving consistently good results from data has produced a high level of interest in the use of sensor monitoring of machining conditions. The research in this thesis uses current technology in calculating cutting conditions, through a combination of Metcut's look up tables and PERA's Macbank data.

### 2.3.3 Setup Planning.

Identifying setup information is an area of work which has received little detailed research. Many researchers, in considering planning problems, have constrained themselves to single setup problems or have used the spindle axis direction, relating to features on the component, as the basis for grouping machining into setups. This ignores the problem of fixturing, the effects of alternative spindle axes directions and tool accessibility problems.

Chan [25], describes a machining process language representing process plans as sequences of setups and machining operations. Setups can be manually described in the language as the positioning and clamping of the workpiece, which is then followed by machining. The need for fixturing in setup determination is described and the automated identification of setups and setup sequences is described as a major research problem, in relating features and operations to produce machining plans.



Armstrong [7] in considering NC code generation from solid modelling, selects setups by considering the material removal capability from each orthogonal approach direction to the workpiece. The maximum material removal direction is taken as the first setup direction, and so on down the list until all material has been removed. This gives no consideration to fixturing and constrains all rough machining in a setup to be machined by a single cutter.

Roy [125] in describing an expert system approach to planning machining sequencing, describes fixturing in terms of resting and clamping points. A knowledge base identifies the required resting and clamping points and an appropriate fixture is selected, after operations have been grouped to maximise the number done in a single setup. This work establishes a framework for an expert system but does not look at detailed interactions between fixtures, operations and cutters.

Chou [32] describes the problem of computerised fixture design as dealing with the arrangement of locators and clamps on the surface of the workpiece. He divides this problem into fixture planning, fixture configuration and fixture construction. The task of fixture planning is to determine the number of setups, and their sequence, and the machining to be done in each setup. He does not, however, address this problem but concentrates on the functional configuration problem. This is to configure a set of location and clamping positions on workpiece surfaces such that the workpiece is totally constrained during machining. Menassa [97] also explores a method for selecting and positioning locating points in fixturing design, using the 3-2-1 locating principle.

Iwata [70] describes the representation of knowledge to identify machining reference surfaces, i.e. the surfaces of the workpiece which must be fixed to the machine table, concentrating on the use of semantic networks, frames and production rules. This work does not consider the interaction of fixtures with machining



operations or cutters, but relies on the rough profile of the part as a means to providing information for decision making.

Darvishi [40] describes an exploratory approach to the design of fixtures using an expert system. This work concentrates on a representation of the facilities required, such as component, machine tool, fixtures and surfaces.

Halvei [56] discusses the influence of tolerances on fixturing, with particular emphasis on the need to re-tolerance dimensions, in some cases, once a fixture datum has been identified. He also describes an algorithm to describe the required dimensions of fixtures to resist cutting forces.

Cutkosky [39], identifies the need for automatic fixture planning in concurrent product and process design. He explores an object oriented system which reasons about friction, to check whether parts will slip and to help to specify the clamping forces required.

Hinduja [63] describes a system called SETUP, which automatically determines work holding parameters for turned components. These parameters are the holding method, the number of setups required and the clamping positions to be used. He uses an analysis of the components technological requirements, against the clamping methods available, to generate a solution.

Boerma [17,18] describes a methodology (FIXES) for the automatic selection of setups and the design of fixtures, based on spindle axis orientation of features and the analysis of their tolerance relationships. Features for positioning, clamping and supporting the workpiece are found, although consideration is not given to fixture interaction with machining operations or cutter accessibility.

Kanumury [76] describes process planning research, which includes fixture planning, within the Quick Turnaround Cell research at Purdue University [28]. He describes the functions of fixture planning as:

- The selection of appropriate fixture elements.
- the maintenance of these in a database to handle relevant queries.
- to understand the part globally to choose an appropriate fixture scheme.
- analyse geometry to determine the location and orientation of fixtures.
- determine interferences with features in a setup.
- map from the part coordinate to the machine coordinates.
- conduct a force analysis on the basis of selected tools.
- evaluate the resulting fixturing scheme.

His work concentrates on the use of a vice fixturing scheme, highlighting the need to reassess fixturing after cutters have been identified.

## CHAPTER 3

### INTEGRATED DESIGN AND MANUFACTURE SYSTEMS.

#### 3.1 INTRODUCTION

Following the critique of research presented in chapter 2, this chapter provides a view of current issues in the move towards integrating design and manufacture functions. This sets the context for the research of the ISS programme in general, and this research in particular.

#### 3.2 TRENDS TOWARDS INTEGRATION.

In recent years the design task has seen a number of software tools produced to aid designers. These Computer Aided Design (CAD) packages range from relatively simple 2D drafting aids, through systems with parametric capabilities, to 3D modellers providing powerful visualisation capabilities. In the same way, manufacturing has seen a range of Computer Aided Manufacture (CAM) software tools emerge. These include process planning systems, providing a plan retrieval capability, and NC part programming systems to aid programmers in tape preparation. There are also a number of software tools to aid production planning and shop floor production activities.

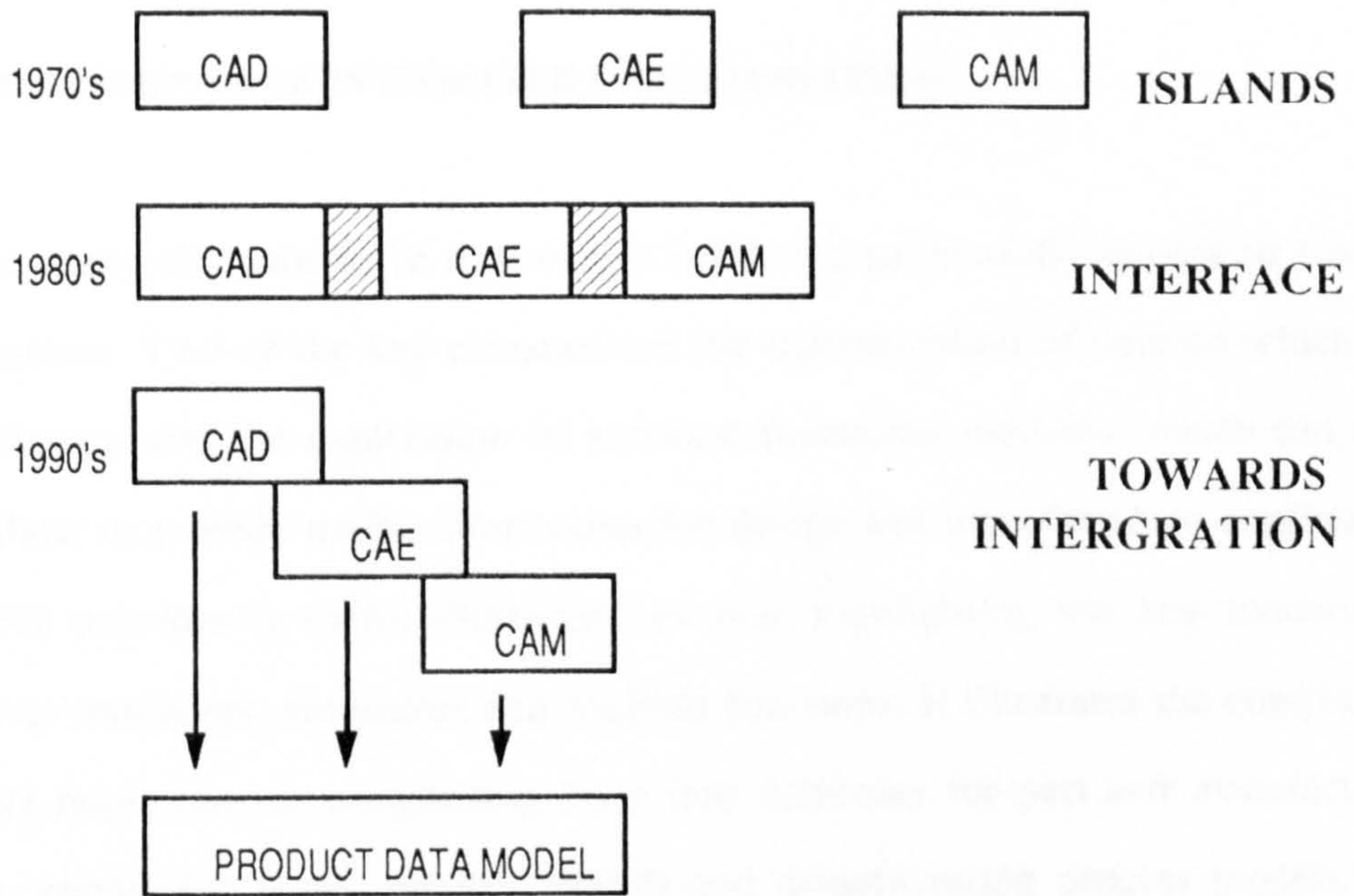


Awareness that many manufacturing functions use engineering drawing information as a major input has led to attempts to bring CAD and CAM software together, in such a way that the CAM software can use data generated through the use of CAD packages. Current commercial software packages, which provide aids to design and manufacture, commonly attack specific problem areas in linking software packages by the provision of an interface. Although there are interfaces between some packages, such as 3D geometry to 2D drafting and 2D drafting to NC graphics, these can be considered as specifically tailored interfaces. This is the case with most software links in manufacturing, leading to "Islands of automation" becoming a commonly used phrase. Attempts to improve the generality of interfacing to CAD packages have been advanced to some extent, with Initial Graphics Exchange System (IGES) attempting to provide a graphics standard, and more recently PDES/STEP [108] attempting to take this forward, to include standards for the description of engineering drawing information as well as shape.

Attempts at providing information from CAD packages to process planning software have been tried with very limited success. The scale of the problem being much greater than simply interfacing. Current CAD/CAM systems rely extensively on the user, with limited information coming from the computing system. This is necessary for two principal reasons: the user's decision making capability is not readily stored in software form, and the information on which the user bases his decisions is wide ranging and needs to interact with the decision making process.

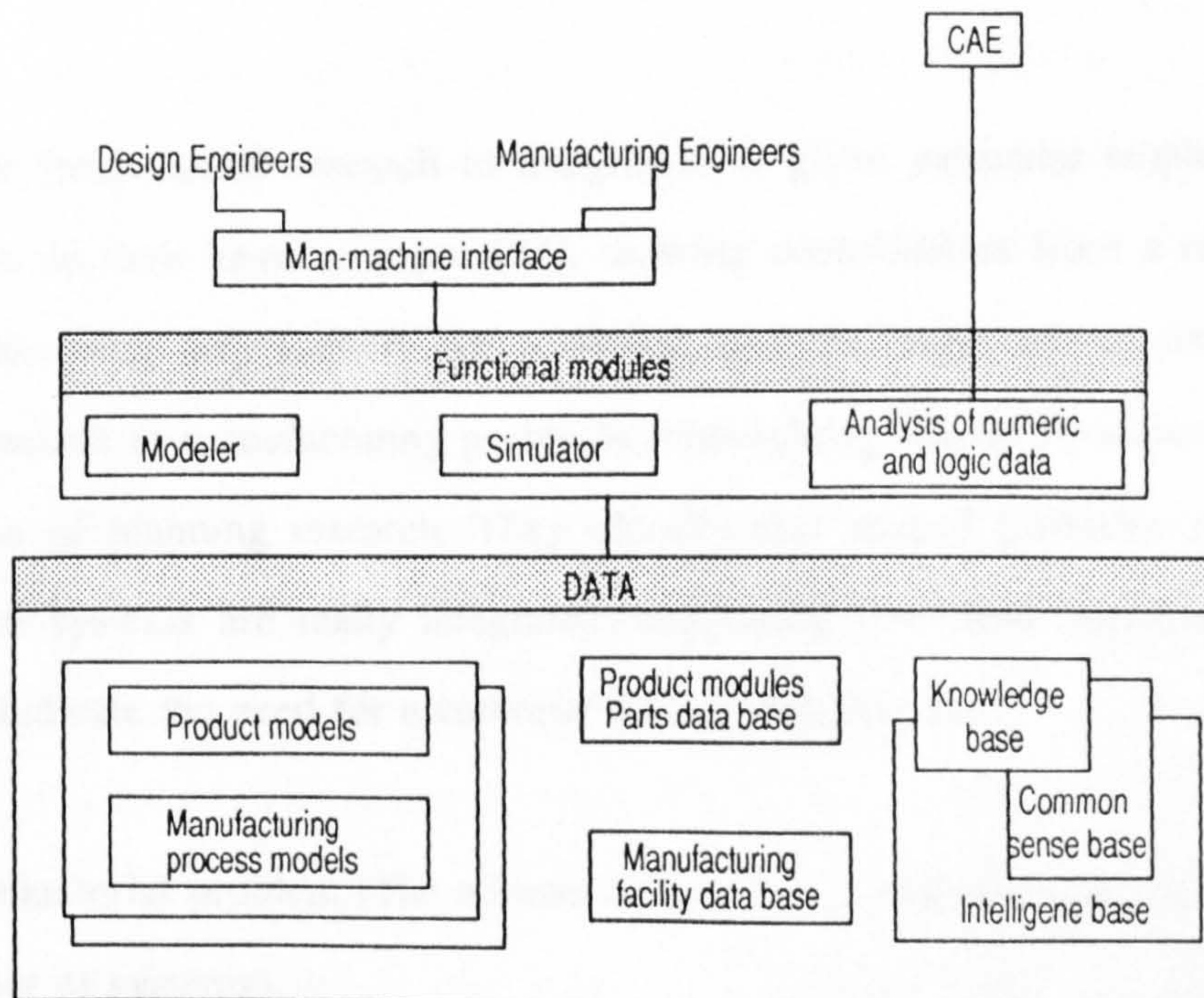
Hence there has been a move from individual software packages in the 1970's, to interfacing systems in the 1980's, to the current aim of integrated systems in the 1990's. A view of these trends, as described by DePennington [46], is illustrated in figure 3.1. A major part of achieving the aim of integration is to establish a common base of product data from which to provide integrated links to design and manufacturing functions.





MECHANICAL ENGINEERING SOLUTIONS [ref. 46]

Figure 3.1



ELEMENTS OF INTEGRATED CAD/CAM SYSTEM [ref. 46]

Figure 3.2



### 3.3 THE ELEMENTS OF INTEGRATED CAD/CAM SYSTEMS

There are many elements to be explored in achieving successfully integrated CAD/CAM systems. Two of the key elements are the representation of data on which to base decisions, and the exploration of software functional modules, which can act on this data, to provide useful information for design and manufacturing engineers. Figure 3.2 provides a useful illustration of this, highlighting the key functional modules as modellers, simulators and analysis functions. It illustrates the complexity of data requirements, categorising them into databases for part and manufacturing data, knowledge bases, product models and manufacturing process models. In the context of this diagram the ISS research programme, in general, and this research in particular, has addressed issues in product modelling, while exploring particular analysis functions, to demonstrate the success of the data representations identified.

The need for fundamental research in integration is given particular emphasis by Ham and Lu. In their keynote paper [57], drawing contributions from a range of experts in the field, they identify the need for, and challenges of, an integrated planning approach to manufacturing problems, highlighting this as an important future direction of planning research. They identify that several problems must be solved before systems are really integrated, suggesting four basic software problems which indicate the need for a common factory database as:

- (i) The combinatorial problem (The number of interfaces explodes with the increasing number of systems).
- (ii) The problem of redundancy and inconsistency (multiple storage of data; different update states).
- (iii) The problem of closed software packages (no access to data structures; algorithms are not available).



(iv) The model problem (different models in different software systems).

By considering a CAD/CAM sub-system they identify that the problems of integration can only be solved by accurate planning of model structures. If the structures of system overlapping models are congruent it is not necessary to store them in a different computer system. This leads to the need for a common product model for the whole company. Therefore they state "The complete modelling of a product, including all necessary information for manufacture, is a basic requirement for the integration of CAD and CAPP".

#### 3.4 PRODUCT MODELLING AS AN AID TO INTEGRATION

In the course of its life a product is conceived, designed, manufactured, maintained, updated and eventually becomes obsolete. Throughout this product life cycle, data concerning the product is both generated and used as the product evolves. Specification data is used by designers who generate design data. In turn the design data is used to initiate the planning of product manufacture, which generates further product data such as process plans and production schedules. This in turn is needed for manufacture which results in further information being generated. Similarly, maintenance schedules for products can be produced and used, with data "from the field" being generated. Data produced at any stage may influence previously generated product data, as well as being used for downstream activities.

The above description of aspects of design and manufacture and their relation to product data, supports the concept of a central product data representation which can provide support for any decision making which relates to a product. This central representation of product data is a Product Model. The importance of product data is that it provides the basis on which decisions can be made, at any stage in a

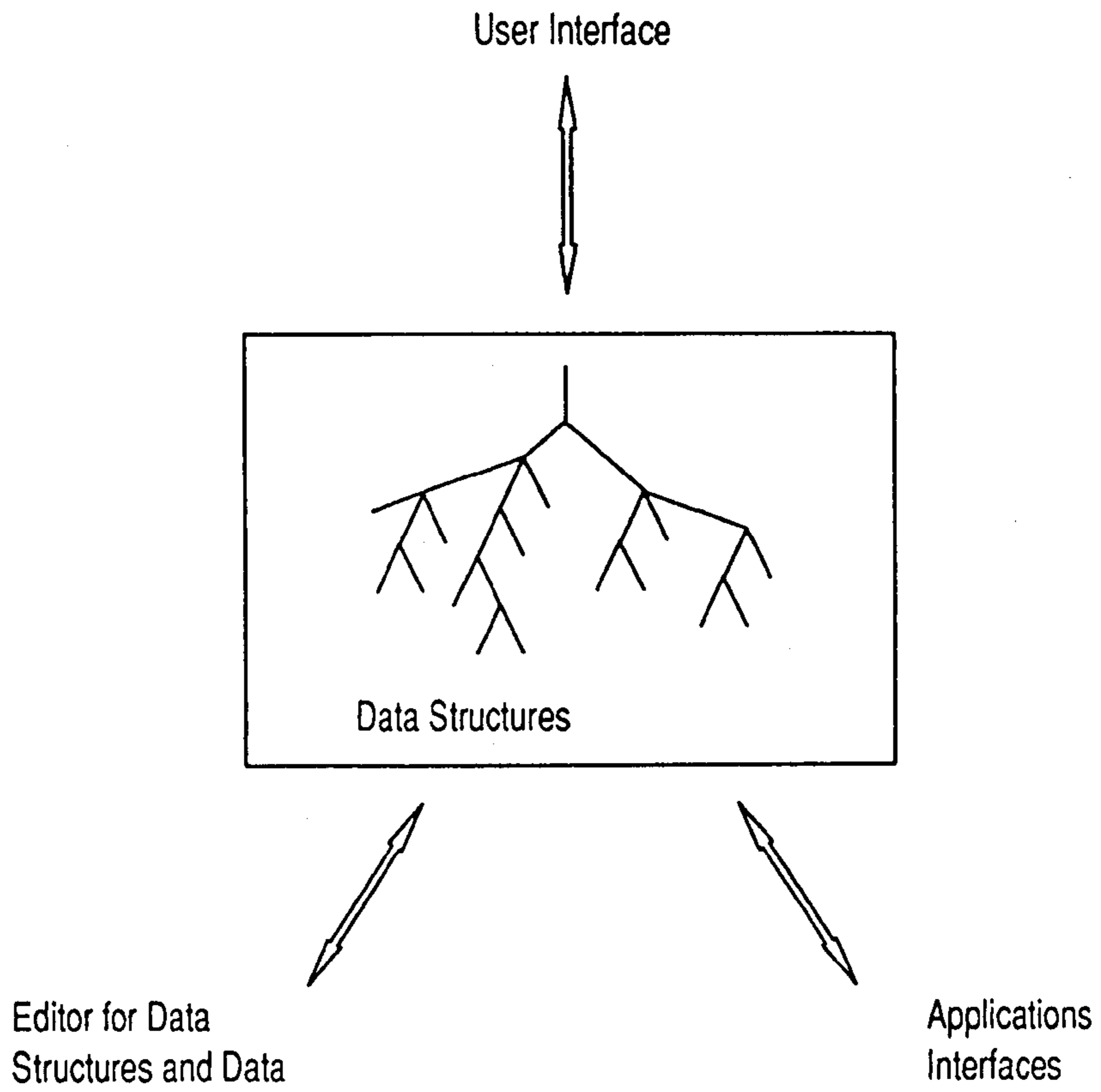
product's life cycle.

As described in the literature survey in chapter 2, a number of researchers have highlighted the importance of representing product data, and an International Standards Organisation committee in the form of PDES/STEP is addressing the problem of providing a standard for product data representation. The representation of product data to provide a basis for integration is a key issue in the ISS programme, and in this research. The results of the work have contributed to the PDES/STEP discussions.

We assume, here, that Product Model representations will be stored in an appropriate database, such that it can be accessed or updated either by a user or by Application software. This implies that three major elements are required in a product modelling system as illustrated in figure 3.3 and listed below:

- (i) an ability to represent and edit information structures in order to build and populate Product Models.
- (ii) an ability to readily interface to Applications software, such that information can be easily passed to and from an Application.
- (iii) an ability to aid a user to visualise and access the product information available.

A prototype software tool, the Structure Editor (SE) [16], provides these capabilities and has been used in the research reported in this thesis, as well as throughout the ISS programme. This has provided the basis to build data representations, and a basis from which to explore the interactions between design and manufacturing Applications which can be achieved through an integrating environment.



**The Structure Editor Facilities.**

**Figure 3.3**



## CHAPTER 4

### THE ISS PROGRAMME - AN EXPLORATION OF INTEGRATION AND AUTOMATION.

#### 4.1 INTRODUCTION.

The previous chapter described the elements involved in integration in design and manufacture and the role of product modelling in achieving that aim. This chapter follows on by describing the nature and scope of the ISS research programme, highlighting the areas of the author's research contribution.

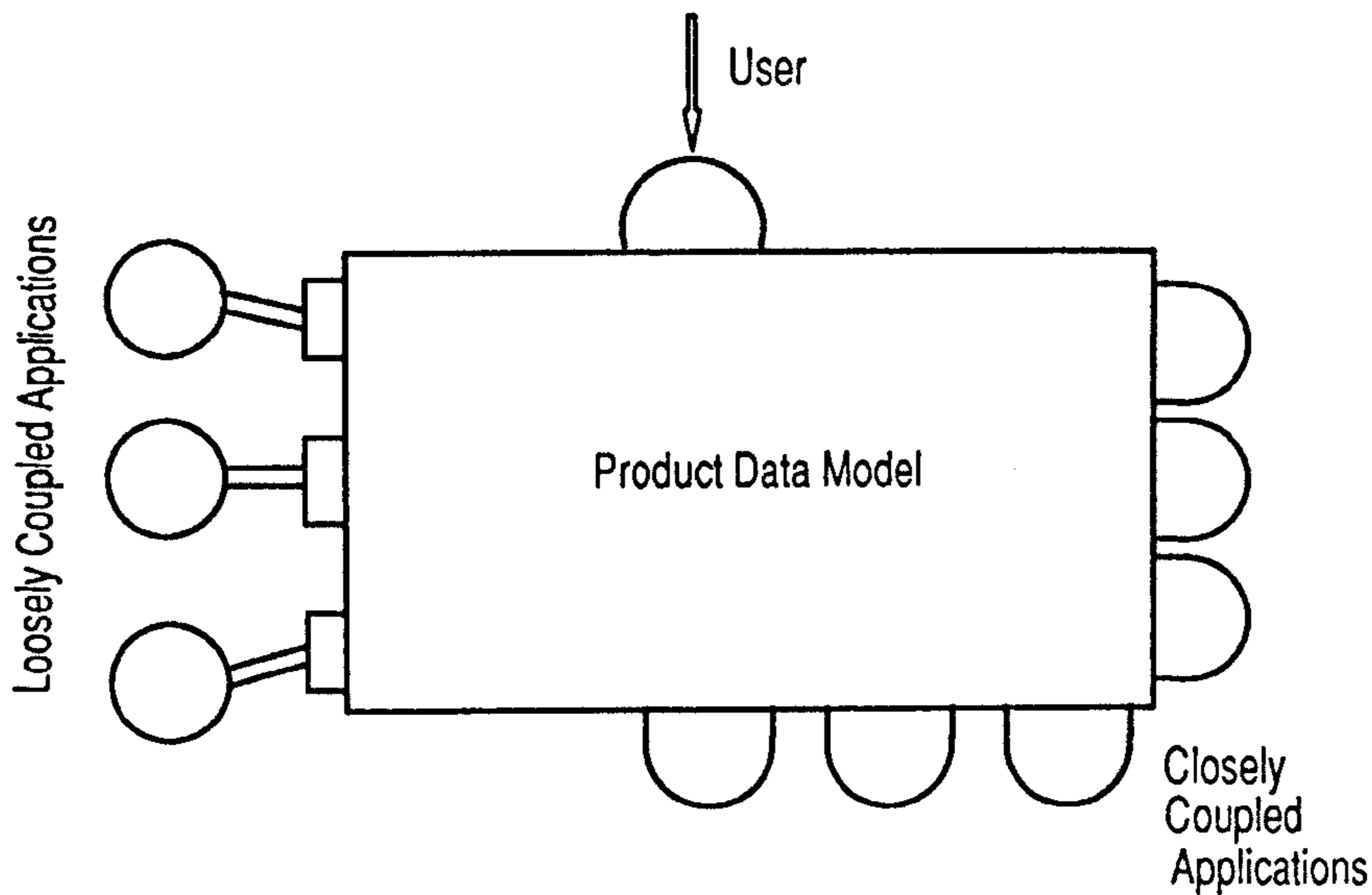
#### 4.2 THE NATURE OF THE ISS PROGRAMME

The ISS research programme followed from earlier collaboration between Leeds and Loughborough Universities which had started to explore the use of solid model data to aid the automation of manufacturing functions. The earlier work quickly identified the need to explore the relationships between a wide range of product related data, in addition to geometry, such as dimensions and tolerances, material specification, manufacturing planning data and feature data. The identification of this requirement, in the earlier programme, resulted in a prototype software tool, called the Structure Editor(SE) [16].

The SE provides an ability to build and edit data structures. These structures can be created from a basic set of building blocks, of which the principal types are called Collections, Selections, Lists and Atoms. A Collection is a group of one or more unlike things. A Selection is a choice from different things. A List is comprised of zero, one or many like things and an Atom is an integer or real number, or a name, a comment or a nil-atom. These terms represent typical software language constructs such as records, enumeration types, linked lists, integers, real numbers and strings as described in the ADA language reference manual [66]. These SE facilities, and the other software utilities, available to the research from the ISS project are described in appendix 1.

Data structures have been built, using the SE, to represent the structure of product models which have then been populated to represent specific products. The structure of a product model has been termed a Product Data Model, while a Product Model is the model of actual data which has been captured in the Product Data Model. These terms, and a framework for product modelling have been determined by members of the ISS research team. A discussion on Product Data Models is provided by Shaw [130]. The terminology defined in the ISS project has been used, where appropriate, throughout this thesis.

The product modelling environment is illustrated in figure 4.1. This shows how Product Data Models take the central role of providing data to Applications software. It also shows how integration issues can be related to "Closely Coupled", or "Loosely Coupled" Applications. Closely Coupled Applications are Applications which are generated with prior knowledge of the representation of product data, while Loosely Coupled Applications are typically third party software items which have no prior knowledge of the product model structure but which, none the less, need to be integrated. Although the ISS programme did research the integration of Loosely Coupled Applications, the research contribution of the author was wholly



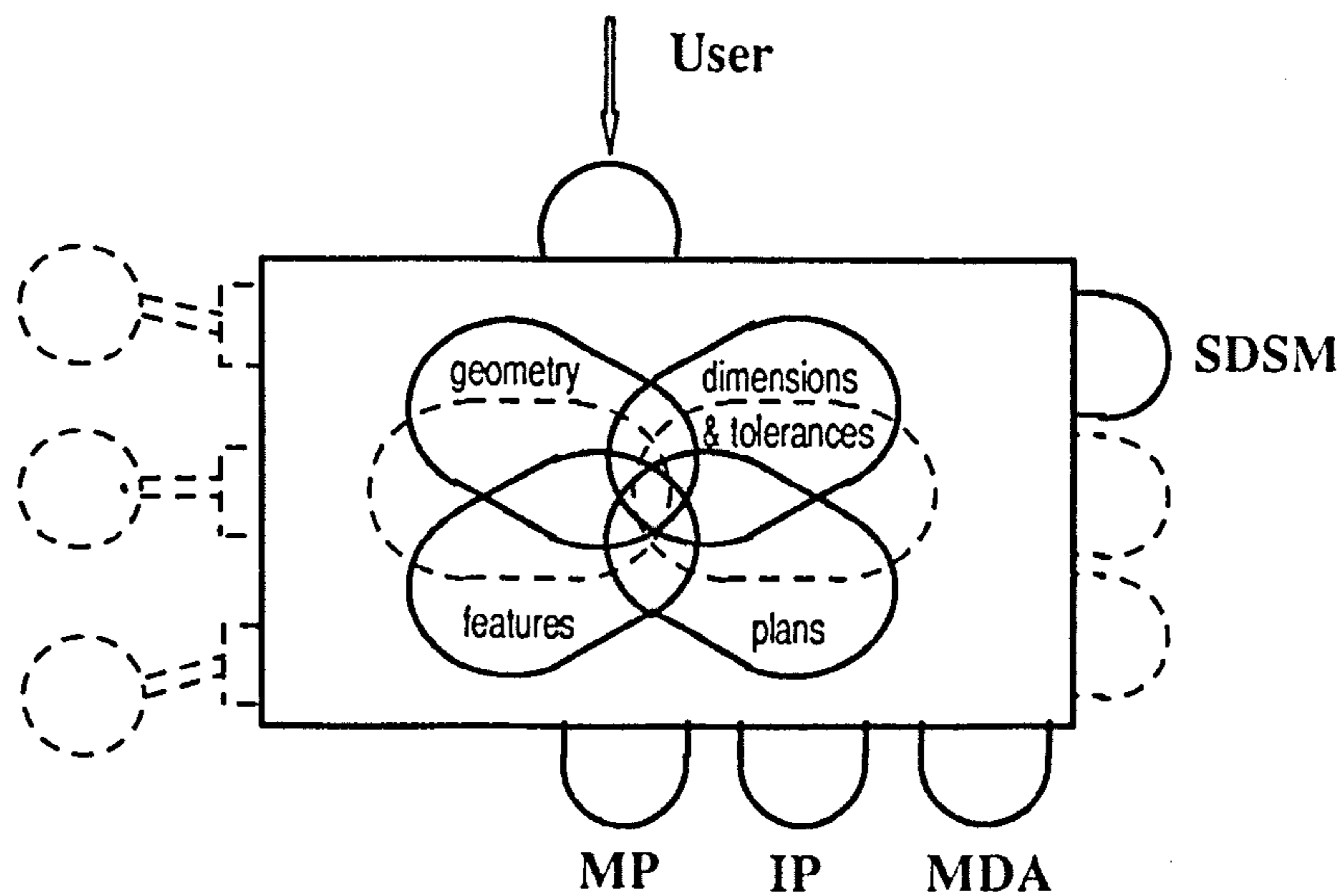
### The Product Modelling Environment

Figure 4.1

concerned with Closely Coupled Applications, as a means to establishing data representations for integration.

The ISS programme has been confined to a narrow "slice" of product data, addressing product models for prismatic components, including geometry representations, dimensions and tolerance representations, feature description and the relationship of process planning data to a product model. These aspects of data representation are illustrated in figure 4.2 which also shows four Closely Coupled Applications. The first three are Machine Planning (MP), Inspection Planning (IP), and Manufacturing Data Analysis (MDA), which form the Manufacturing Code Generation Applications pursued in the ISS programme. The fourth Application is the Spatially Divided Solid Modeller (SDSM), a cell decomposition model used in the research. Its inclusion as an Application highlights that even geometric models become support tools for product modelling, rather than a central source of data themselves.





## The ISS Data Requirements and Applications

Figure 4.2

In its exploration of fast prototyping, the ISS programme aimed to establish a Product Data Model to enable the integration of the MCG Applications. Access to product data leads to the ability to introduce levels of automation into Applications, where adequate data for decision making is available. The use of a product modelling environment also means that the system user can access all product data contained in the model. This enables the user to make changes where Application generated data is inadequate.

### 4.3 AN EXPLORATION INTO THE INTEGRATION AND AUTOMATION OF INTERACTING ACTIVITIES.

#### 4.3.1 Manufacturing Code Generation.

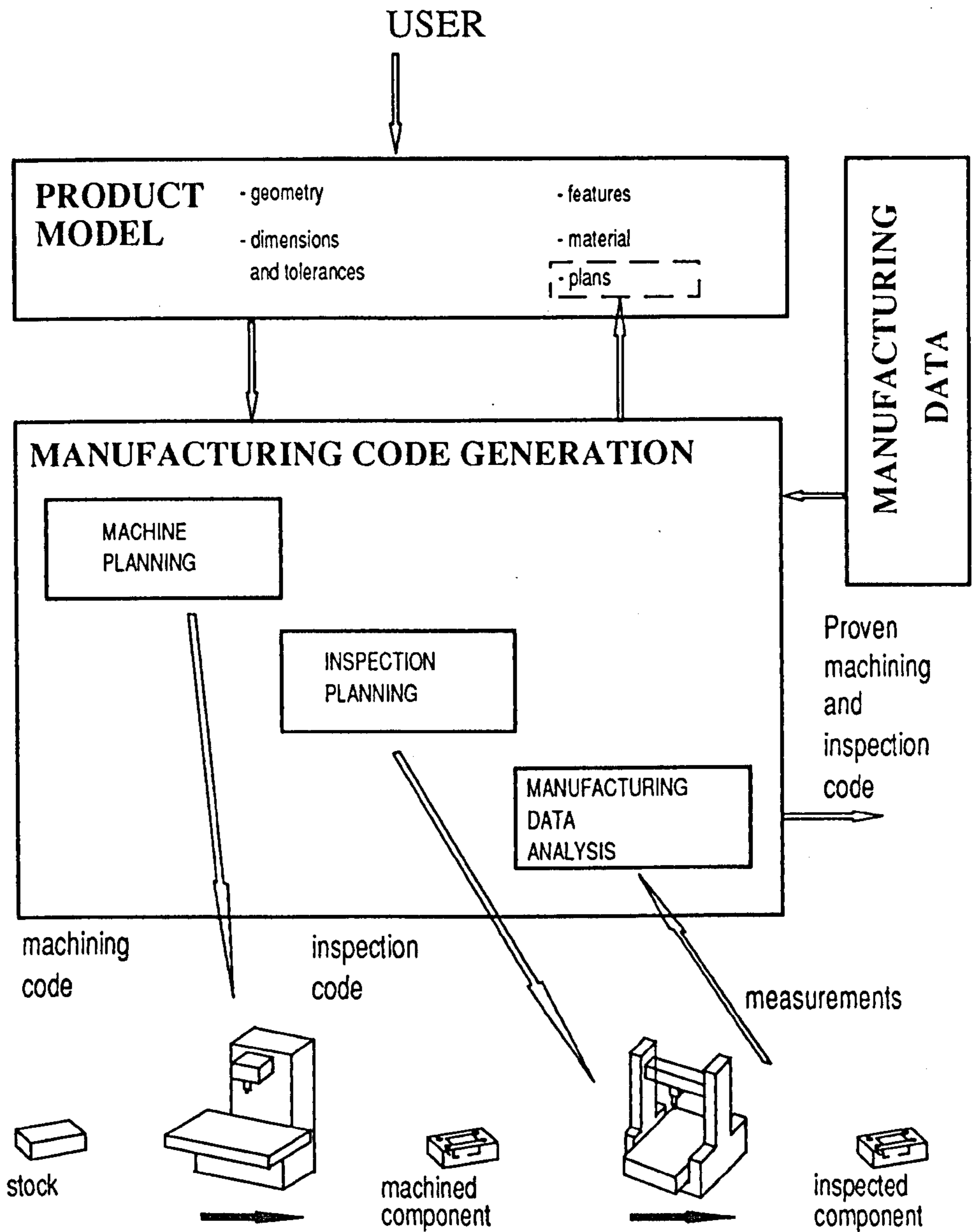
The narrow slice concept of the ISS programme emphasised the requirement for systems integration. This pursuit of integration was based on the use of the product model concept to provide the integrating framework for computer aided design and

manufacture. The manufacturing functions in the programme being termed Manufacturing Code Generation (MCG). Figure 4.3 illustrates these MCG activities and their general links to Product Modelling and to manufacturing to achieve plans for machining and inspection and proven part programs. The following sections provide a brief description of each of Machine Planning, Inspection Planning and Manufacturing Data Analysis, highlighting the role of the author's research.

### 4.3.2 Manufacturing Data Analysis

The Manufacturing Data Analysis work was concerned with the data and methods requirements for the analysis of machined and inspected components and the feedback of error correction data. This was the responsibility of L. Lee and is reported in [82].

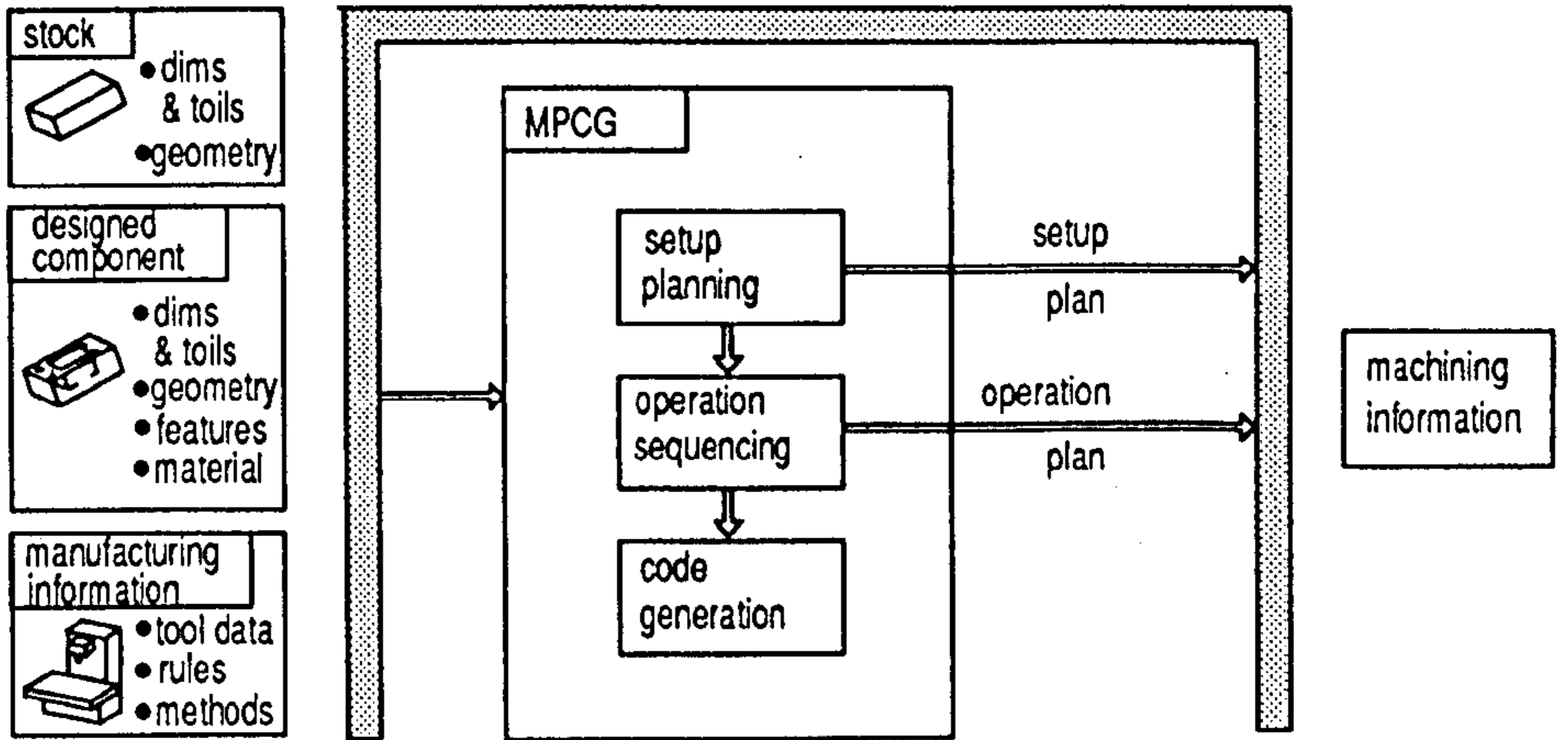
This work considered workpiece measurement analysis, fault analysis and manufacturing process analysis to provide data correction as illustrated in figure 4.4(c). The significant aspects of product data required for this work are highlighted as the dimensions and tolerances of the designed component, measurements of the actual component and manufacturing information.



**The Integration of Manufacturing Code Generation  
in a Product Model Environment**

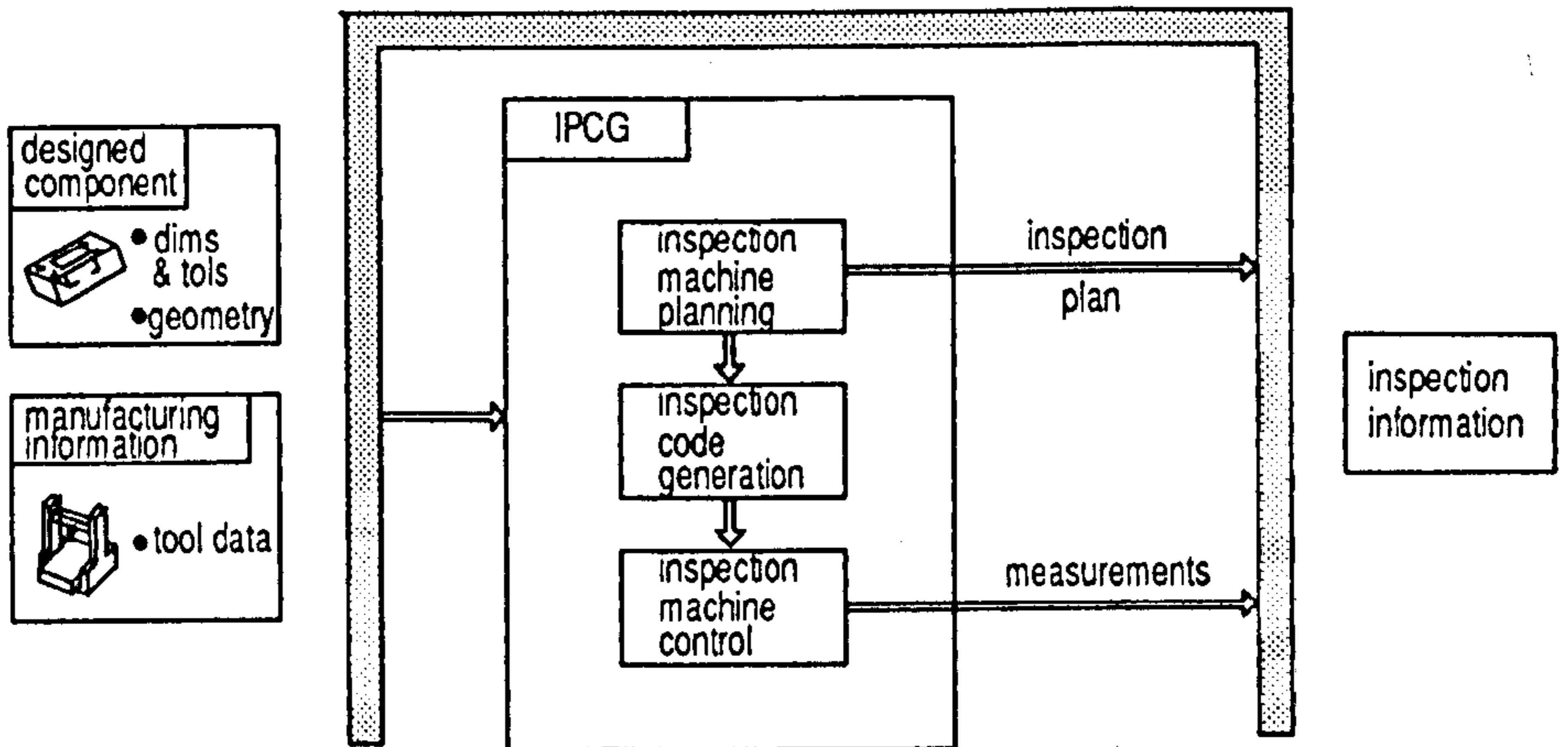
**Figure 4.3**





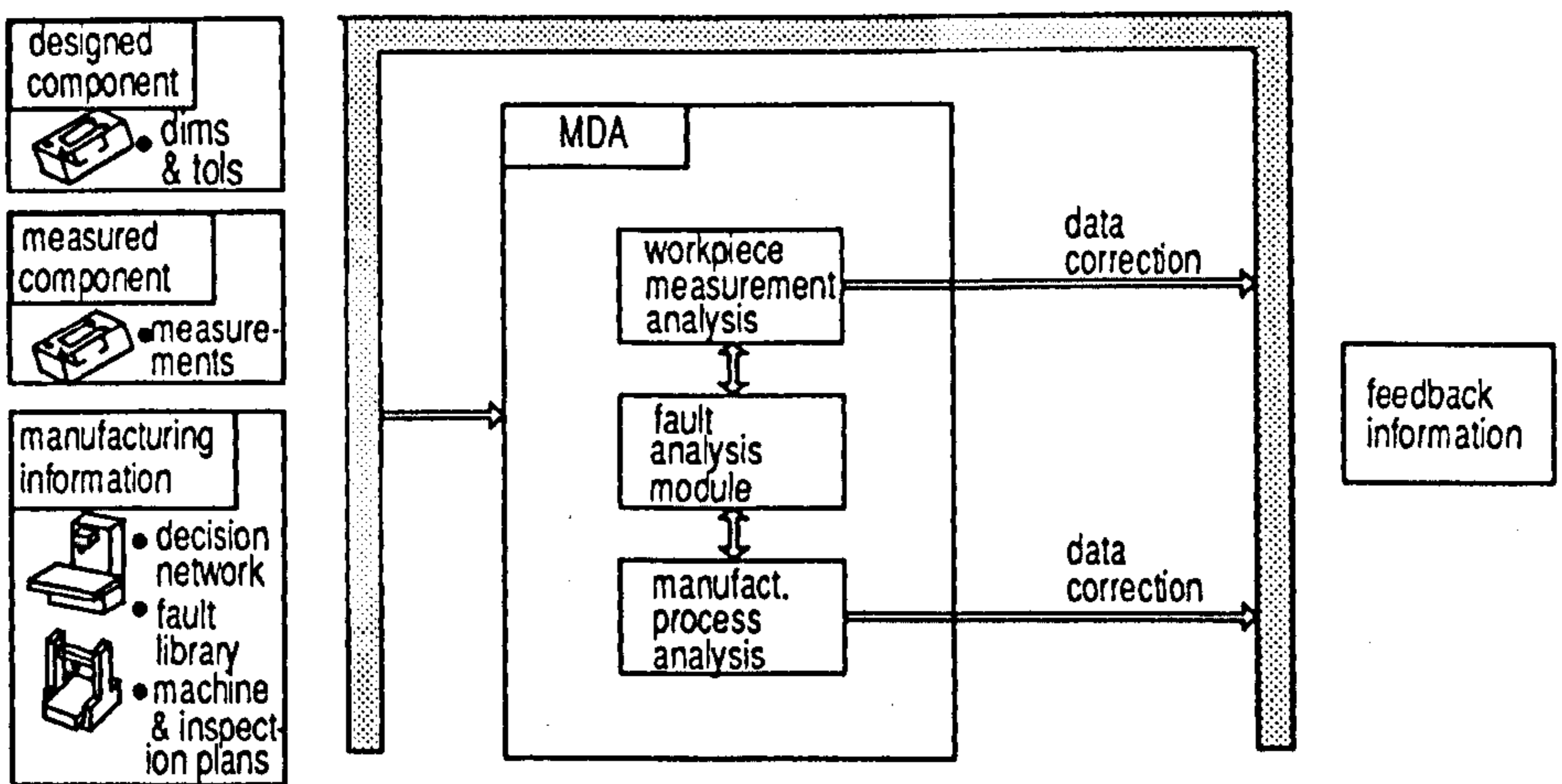
**MACHINE PLAN & CODE GENERATION**

(a)



**INSPECTION PLAN & CODE GENERATION**

(b)



**MANUFACTURING DATA ANALYSIS**

(c)

Figure 4.4

### 4.3.3 Inspection Planning

The Planning for Inspection work was concerned with the data and methods required to plan the use of a coordinate measuring machine. This was the responsibility of M.J. Corrigall and is reported in [35].

The work considered inspection machine planning, inspection code generation and inspection machine control, to provide inspection plans and measurements which could be used in MDA. This is illustrated in figure 4.4(b). The significant data required for this work was identified as the dimensions and tolerances and the geometry of the designed component.

### 4.3.4 Machine Planning

The research group took a view on Machine Planning which influenced the boundaries of the author's work. The broader issues of process planning and the use of artificial intelligence techniques were being pursued in the parallel, Alvey DtoP, programme. These issues were, therefore, not addressed by the author. Work within the ISS programme on the analysis of complex geometry to identify machining trajectories was the responsibility of P. Bell [12]. Bell explored computationally effective algorithms to achieve the automatic generation of NC code for the general case of machining three dimensional forms; this concentrated on rough machining with particular reference to the exploitation of the ISS geometric tools.

The capture of product data, which enables a range of Applications to interact, opens up future possibilities in the production of more effective software tools to aid design and manufacture. It is proposed that the Product Data Model provides the key to enabling this interaction. The author took responsibility for the research



into a Closely Coupled Machine Planner, exploring not only the data which would enable Machine Planning to be performed, but also contributing to a PDM which would allow interactions between the range of Applications being addressed in the ISS programme.

The data requirements for a Machine Planner, as illustrated in figure 4.4(a), were identified as the stock geometry and dimensional data, the designed component geometry, features, and dimensional data as well as manufacturing information. The functions within MP which were addressed by the author, as a means to exploring integration, were setup planning, operation sequencing and NC code generation, to produce machining information directly related to a component. The exploration of the planning data to be associated with features and components to enable the Close Coupling of a Machine Planner formed a key part of the research, along with the exploration of links to geometric, dimensional, and the other MCG Applications, through the PDM.

Ham [57] identifies that production, process and operation planning are highly interrelated in practice, even though each has its own distinctive goal and scope. These three should therefore be considered as integral components rather than separate pieces of a complete manufacturing planning system. and should not be studied in isolation. How then can an individual researcher, or even a group of researchers, make a valuable contribution to an area of research which must draw on such a wide range of manufacturing and research understanding? It is the view of the author that it is important that each piece of research undertaken must be performed within the context of a defined set of relationships between manufacturing activities. Hence it is important to be able to model manufacturing activities and their relationships in a formal manner.



To provide clear interaction links between the planning activities used in the research exploration of integration, the IDEF0 activity modelling technique [123] has been used. This provided the ISS project members with a formal method of exploring the relationships between the design and manufacturing activities being addressed, and their context in the broader scope of producing a product. Hence it provided the starting point in the exploration of the interactions which the PDM would have to be able to support, if integration were to be achieved.

The recently started ESPRIT IMPACT programme [68], addressing the integration of design, operation planning and NC code generation, is also taking the view that information modelling is the key to the integration and automation of design and manufacture processes. This work is also exploring the use of IDEF0 for activity modelling and is exploring the use of EXPRESS for data modelling.

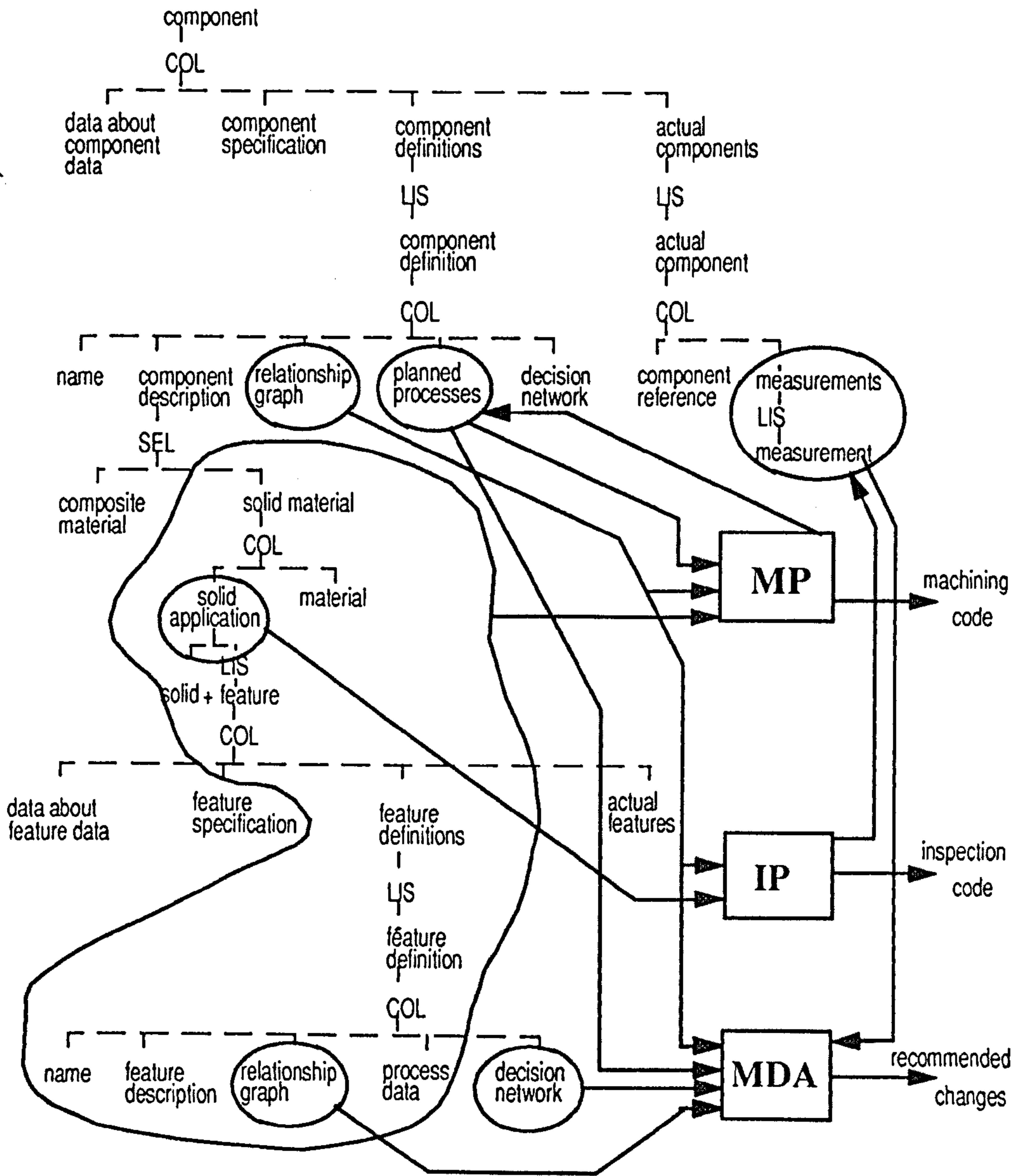
Although product data is a key input to planning decision making and to integration, a further important input is the capability of the machines which are under consideration. The differences in capability can range from slight, e.g. when comparing three and four axis machining centres, to large, e.g. when comparing three axis machining centres, to coordinate measuring machines. Even though the description differences may be slight, the effects of these differences on planning decisions may be significant. As such, the view taken by the author is that a Machine Planner should exist, for each machine type, to plan machine level decisions. In this way, it is suggested, a range of Machine Planners could be used by a process planner to simulate alternative machining possibilities. The initial part of the author's research, addressing a structure for Machine Planning, has been published in the International Journal of Production Research [14].

### 4.3.5 Product Data Flows in Manufacturing Code Generation.

The parallel pursuit of the three Manufacturing Code Generation functions in the ISS programme allowed the research team to explore three different viewpoints on data requirements. This enabled the data requirements for the integration of all three Applications to be explored in addition to the integration issues within each sub-topic. The author made a significant contribution to the exploration of the interaction of these manufacturing functions, to ensure that an understanding of the common data requirements, and the reasons for differences in data requirements, was achieved.

Figure 4.5 provides an illustration of the MCG Applications and the flows of product data which are required to enable the generation of machining code, inspection code and recommended changes. This highlights the use of a common source of product data for each of MP, IP and MDA. The product data aspects of the figure will be discussed in the next chapter. By choosing to concentrate on integration as the main theme, the research was able to explore the relationships between geometry, dimensions and tolerances, features, planning data and measurements of components, to generate manufacturing code in a truly integrated design and manufacturing environment.

The key aspects of product data of relevance to the author's work are described in the following section, while the author's contribution to the provision of a representation of data to enable integrated MCG is described in the following chapter.



Product Data Flows in MCG

Figure 4.5



#### 4.4 PRODUCT DATA TO ENABLE MANUFACTURING CODE GENERATION.

##### 4.4.1 Workpiece Geometry and Feature Data.

It is well recognised, as described in the literature survey, that it is important to have an unambiguous representation of geometry when addressing links from geometry to planning Applications. Such a representation can be provided by a solid model. The representation of geometry is not an issue in this research, but rather the problem of integrating the required geometric data into a Machine Planner. This has involved the identification of the geometric data, needed in planning, and the exploration of the required links to the ISS geometric tools, to provide the necessary data.

The primary geometric data representation mechanism in the PDM was a Constructive Solid Geometry (CSG) Modeller, based on half space definitions. This representation allowed geometric models of features and components to be build in the Product Model environment. A separate geometric Application was available to extract the geometric model data and construct a CSG tree. In addition to the CSG representation a second geometric Application was available which could build an SDSM representation from a CSG tree. The combination of these two geometric representations proved particularly valuable in exploring feature interaction data, as described in chapter 7. A description of the SDSM is provided in appendix 1.

A feature can be considered to be a piece of geometry which has some significance for manufacture. The understanding and use of features is an area of research which has received considerable attention from the research community, as described in the literature survey. One of the problems with features is that the

significance of a shape is dependent on the particular manufacturing viewpoint being taken. For example, design features are likely to be different from assembly features, which, in turn, are likely to be different from machining features. By exploring Machine Planning, Inspection Planning and Manufacturing Data Analysis, the ISS programme has explored three views of the manufacturing significance of features to identify how these should be captured in a product model representation. The author's research has established a data representation to be associated with features for machining while contributing to comparative explorations of features for Inspection Planning and Manufacturing Data Analysis.

In general, research into features is split into two approaches. The first is to design a solid in terms of features, termed "design by features". The second is the redefinition of a solid in terms of features, termed "feature recognition". The first of these methods restricts the designer to the use of pre-defined features, which can limit the complexity of design which can be represented. The second method avoids this restriction by attempting to find features on an already designed component description.

Feature recognition is a complex problem which is only now beginning to progress beyond the recognition of the simplest geometric shapes. Recently the PART system of feature recognition [140] has shown that combinations of simple shapes, as compound features, can be recognized. However interactions between features is still a problem. Willis [149] states that features in isolation are insufficient for the synthesis of a machining plan for the whole component, identifying the need to consider the machining of thin floors, machining through thin floors and machining thin walls. Kanumury [76] also identifies the need to consider interactions between fixtures and machining features.



Set in the context of the ISS programme, the author's work has considered machining, fixture and datum features. The research related to machining features, was restricted to the exploration of a structured representation of feature process data and its association with geometry, as a required input to a Closely Coupled Machine Planner. In addition, data on feature interactions, as required by a Machine Planner, have been explored, by linking, through the PDM, to the SDSM. This has enabled the author to demonstrate how feature process data and feature interactions can be captured to aid the integration of a Machine Planner in a Product Model Environment.

#### **4.4.2 Associating Dimensions and Tolerances with Solid Geometry.**

Dimensions and tolerances are an essential part of engineering design data. Appropriate representations, linked to solid models, have not been available, and major research effort is still ongoing. An international workshop sponsored by the National Science Foundation [139] states that a comprehensive theoretical basis for developing computer representations of toleranced parts and assemblies does not exist and, therefore, major research is required to establish sound mathematical foundations on which to build computer representations.

Researching dimensional links to Product Models, in the ISS project, has been the responsibility of L. P. Wickens, whose work has resulted in a Relationship Graph, reported in Wickens [148]. His work provides a means of linking dimensions and tolerances to the relationships between the faces of a solid model. Corrigan [35] has shown how the relationship graph, in conjunction with geometry, can be used in inspection planning, while Lee [82] has shown, within a limited scope, how the relationship graph can be used in Manufacturing Data Analysis. The lack of availability of a linked description of geometry and dimensional data, at the time when the author's work was being performed, has hampered progress in producing in-



tegrated links to the Machine Planner. There is still a need to provide a combined link from the Relationship Graph to feature data, as opposed to a general geometric description. The work of the author shows how the Relationship Graph, in conjunction with geometry and feature data, could be used to provide both feature specific dimensional data and feature interaction data for Machine Planning. This is described in chapter 8.

#### **4.4.3 Manufacturing Planning Data.**

The data requirements for planning are often considered to be purely geometry, material, dimensions and tolerances, with the features of significance on the component being the final link. For example the PART system takes this view [140]. The author has taken the view that there is a need for an intermediate planning data structure to be associated with a component model. This then provides a basis for the capture of planning data and a means by which planning Applications can interact.

The author's research, within the ISS programme, has contributed to a novel concept showing how MCG can be integrated through a Product Data Model, producing some radically new results which demonstrate how the interaction of Applications can be achieved. This makes it possible to anticipate a structure through which product data can flow between Applications, hence effecting a comprehensive bridge between design and manufacture i.e. a truly integrated CAD/CAM system.

## CHAPTER 5

### INTEGRATING MANUFACTURING CODE GENERATION IN A PRODUCT MODEL ENVIRONMENT.

#### 5.1 INTRODUCTION.

This chapter describes the research performed, in conjunction with co-researchers, to identify the data requirements of Machine Planning, Inspection Planning and Manufacturing Data Analysis. In particular, the work performed to identify the data requirements which are common between these three functions are described.

The importance of capturing activity interactions is discussed and the method used to model these is described. The way in which common workpieces have been used to aid the definition of data requirements is described, as is the Product Data Model which resulted from the combined research.

#### 5.2 CAPTURING THE INTERACTION BETWEEN ACTIVITIES

##### 5.2.1 The Importance of Understanding Activity Interactions.

To achieve an environment which will enable integration, Product Data Models must be accurately planned, such that the resulting data representation provides for the storage of all necessary information for manufacture. This is a requirement for successful integration, as identified by Ham [57]. The planning of these data struc-

tures is dependent on the understanding of the data requirements and data flows between the manufacturing functions, which are to use the data structure. Hence an accurately planned Product Data Model depends on a clear and common understanding of the functions which are to be integrated through its use.

### 5.2.2 Problems in Defining Interactions.

To identify a clearly defined representation of the data flows between manufacturing activities is not straight forward. There are many functions to be performed in design and manufacture and their relationships are complex. Again, Ham and Lu have identified the need for a logical approach to the identification of a planning structure, if integration and automation are to be achieved in future computer aided process planning systems. Also Smith [132] describes the need, in computer integrated software development, to firstly identify the essential tasks, or activities, and their interrelationships and, secondly, to create the data models which contain the basic data to provide support for the tasks.

The relationships between manufacturing activities are not well understood and, as data is not always available on which to base decisions, assumptions are often made by design and planning engineers which, when incorrectly made, can have expensive consequences [142]. This has led to a drive towards achieving simultaneous engineering, where all data and knowledge related to a problem are brought together concurrently to achieve a solution. Design and planning engineers, working together, help to overcome some of the problems of making incorrect assumptions, by bringing together the necessary knowledge to tackle a problem. Hence knowledge of how activities are performed, as well as data, provides an input to the performance of a manufacturing activity. However, there is still a need to use a structured approach to design and manufacture, such that the data required at each stage in the process can be made available.



To achieve such an understanding of the interrelationships between design and manufacturing functions requires some formal basis for discussion. Such a basis can lead to the evolution of a greater understanding of the ways in which design and manufacturing functions can be represented by software Applications. This, in turn, provides a basis on which Product Data Models can be defined.

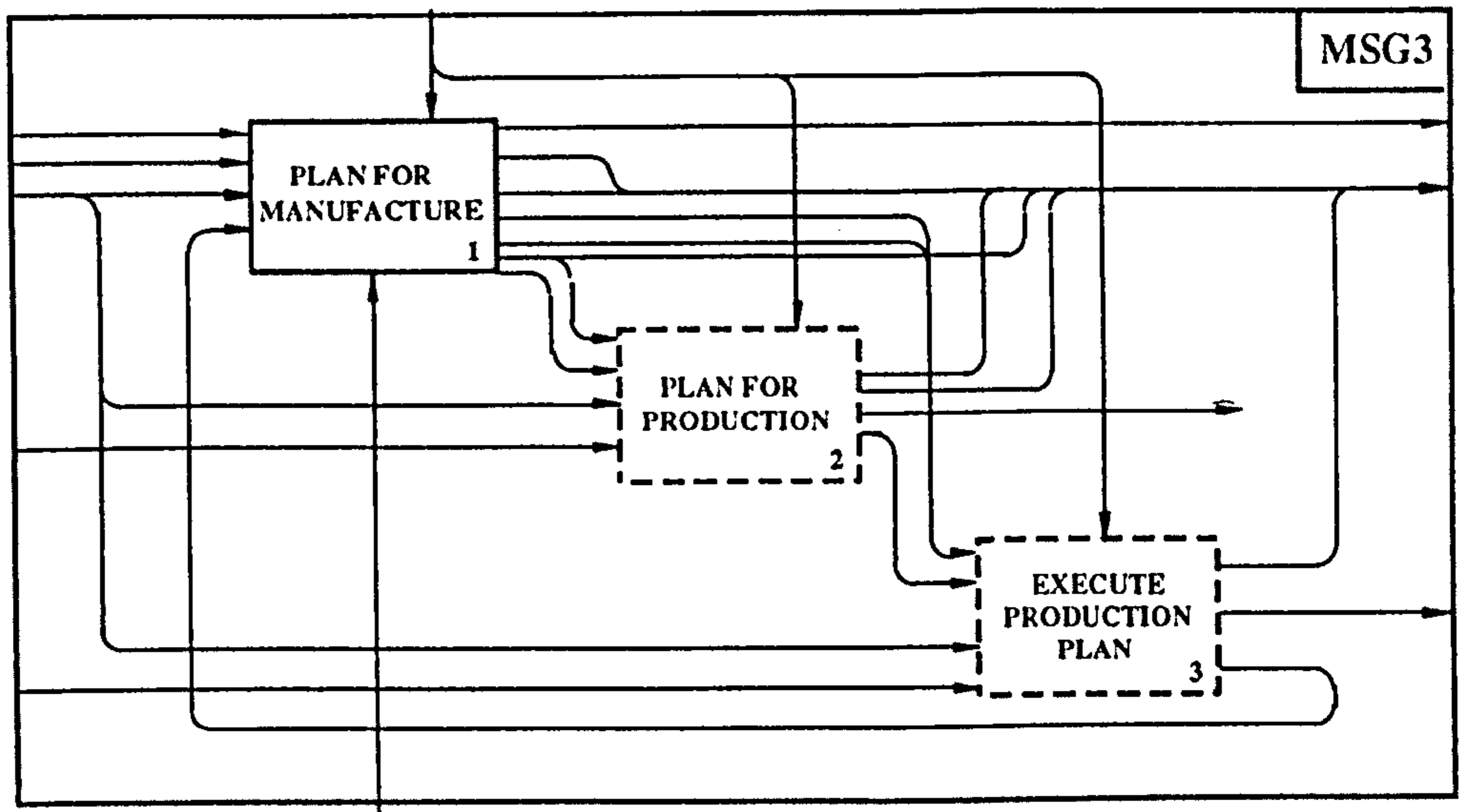
### **5.2.3 Capturing the Interactions Involved in Producing a Product.**

Following from the above requirement to achieve an understanding of the interrelationships between design and manufacturing functions, the research group pursued the use of the IDEF0 modelling methodology. This had the aim of setting the work of ISS, and this research in particular, into a broader manufacturing context, as well as providing an activity model on which to start the exploration of Product Data Models. The resulting IDEF0 model of *Produce a Product* is provided in appendix 2.

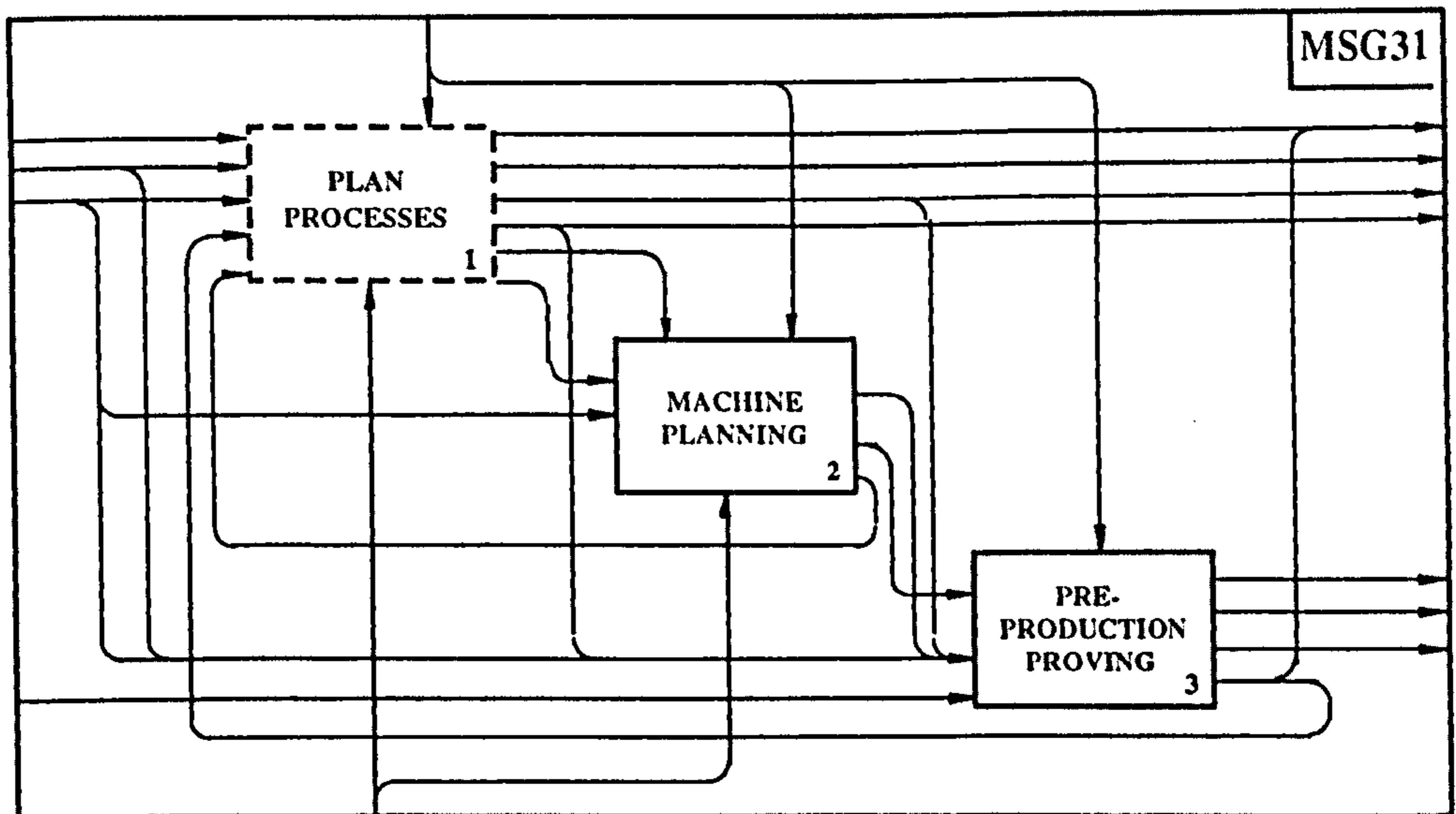
The generation of the IDEF0 model took many iterations and involved frequent discussions between the research group members and industrial collaborators before a generally agreed model could be achieved. It is interesting to note that a change in the viewpoint, or emphasis, by a reader can lead to a significantly different model being generated. For example, there is a strong argument, in simultaneous engineering, to include planning for manufacture as part of the design activity, which would not necessarily be the case in other circumstances. It is the author's view that it is unlikely that a generic model will ever be produced as many different viewpoints will always exist. However, this does not negate the value of the model as it provides a strong basis from which to gain an understanding of activity interactions, and even to gain some insight into the different viewpoints which exist. This understanding enables areas of interaction to be defined in greater detail and with greater clarity.

It is also interesting to note that two other IDEF0 models produced since the ISS model, and completely independently, one for CAM-I by Ferriera [47] and one by Colquhoun [34], do show similarities with our model. The CAM-I model concentrates on automated process planning and identifies part understanding, process selection and ordering, machine tool selection, setup planning, operation planning, and evaluation and documentation as the key tasks of interest to them. The Colquhoun model is also a model of process planning, but it recognises that process planning is an integral part of planning for manufacture in the wider environment of producing new products. Their breakdown of process planning concentrates on a Variant approach, identifying similar components in order to produce plans for new components. It is the author's view that the differences in the models are principally due to the differences in viewpoint taken. These differences suggest that future work aimed at providing generic solutions for computer integrated manufacturing must address the effect of viewpoint in their research.

The ISS model concentrated on showing the MCG activity interrelationships in the context of producing a product and the three key elements of the total model are illustrated in figure 5.1. *Planning for Manufacture* is a key activity in the model with figure 5.1 (a) showing the breakdown of the *Manufacture* activity. This breakdown illustrates how inputs from *Plan Production* and *Execute Production Plan* impinge on *Plan for Manufacture Decisions*.



(a) MANUFACTURE



(b) PLAN FOR MANUFACTURE

Figure 5.1



The *Plan for Manufacture* activity encompasses typical process planning tasks, part programming and pre-production proving functions. Machine Planning falls within the *Plan for Manufacture* activity as shown in figure 5.1 (b). The *Machine Planning* activity takes data, concerning the workpiece to be machined, as input from the *Plan Processes* activity, and it is controlled by the type of machine being considered, and knowledge of how the machine is used. *Machine Planning* then outputs machine plans which can be used for process planning and part programs which can be sent for proving. The *Machine Planning* activity, as illustrated in figure 5.1 (b), can in principle encompass any machine type. In the ISS project, two machine types were considered, a 3 axis vertical machining centre and a coordinate measuring machine. The former has continued to be termed Machine Planning while the latter, to avoid confusion, has been termed Inspection Planning.

The *Pre-production Proving* activity ensures that only proven part programs are used when executing a production run. The Manufacturing Data Analysis research of ISS forms a part of this activity. Components produced from the original machining instructions and part programs are checked and recommendations made as to how errors found can be corrected. The cycle is repeated until satisfactory components are produced and the part programs are proven.

The activity models produced, provided general data flow requirements which aided the identification of the general structure for the relevant product data. The data definition at this level, however, is still very weak, and open to alternative interpretations. Significantly more detail was therefore required to fully define the data requirements. This further requirement was pursued in two ways: firstly by performing more detailed activity modelling on each of Machine Planning, Inspection Planning and Manufacturing Data Analysis; secondly by focusing on particular workpiece examples, to identify more directly, their specific data requirements.

### **5.3 IDENTIFYING THE PRODUCT DATA REQUIREMENTS FOR MCG**

#### **5.3.1 Maintaining a Focus on Integration Through the Use of a Common Workpiece.**

As explained earlier, a central aim of the ISS research was to establish product data structures to enable integration of MCG Applications, with the author working on the Machine Planning aspect of this. IDEF0 modelling provides some clarification of data flows, but does not provide a well defined data definition. The challenge of identifying a central product data model introduced many interacting issues which were not easy to solve. Those of relevance to the author's work were:

- What inputs are needed for, and outputs generated by, Machine Planning, Inspection Planning and Manufacturing Data Analysis, and how do they relate?
- What is the manufacturing significance of features for Machine Planning, Inspection Planning and Manufacturing Data Analysis? What differences in significance exist and why?
- How can dimension and tolerance data be related to geometry and used by MCG Applications?
- How can feature interaction data be found?
- What relationship should there be between feature pre-definition and feature recognition?
- What data representation should be used such that setup, operation and NC planning can be integrated in a Product Model environment?



To bring some understanding to these problem areas, and to maintain a common level of understanding amongst the research team members, the research group chose to focus on a common workpiece. This was initially a very simple workpiece, into which increasing levels of complexity were introduced as the level of understanding of the underlying data requirements evolved. Initially Pockets, Slots, Channels, Holes, Faces and Bosses were considered. Then combinations of these shapes within a setup were considered. Finally combinations of these shapes in multi-setup components were considered. The detailed exploration of MP integration, described in the following chapters, has led to the belief, that future integrated systems, offer the potential to exploit greater levels of interaction between Application software, thereby enabling more comprehensive software aids for design and manufacture to be produced.

### **5.3.2 A Summary of Data Requirements for MCG.**

The data requirements for each of MP, IP and MDA are illustrated in figure 4.5. This figure highlights that the only common data requirements between all three activities are the dimensions and tolerances, represented in the Relationship Graph, of the designed component. The planning activities both require the geometry of the designed component while MP also requires the stock geometry, dimensions and tolerances.

The use of features in each of the Applications of MP, IP and MDA have been investigated in the ISS project, by the researchers involved in each area i.e. M. Corrigan, L.Lee and the author. From this it has been found that features are very much Application related. Features for Machine Planning and for Manufacturing Data Analysis are similar, while features for Inspection Planning bear no particular relationship to the other two.



Machining features in Machine Planning, and Manufacturing Data Analysis features, are both concerned with geometric surfaces, which are grouped to relate to machining methods. As both Applications are interested in machining methods, the shapes of interest to both are therefore the same. However MP requires data which can result in the identification of machining methods, while MDA requires data which will result in the identification of the probable causes of errors. The significant data related to shape is therefore different for the two Applications.

In contrast to MP and MDA, Inspection Planning is concerned with surface pairs, rather than groups of surfaces, and the significant data for IP are simply the dimensions and tolerances relating the surfaces. There is no need, therefore, to associate additional feature data with a product definition in the case of Inspection Planning.

Some product data inputs to Applications are the data outputs of others. MDA requires measurements which are produced by IP and also requires machining and inspection plans produced by MP and IP. In addition to product data, all three Applications have a requirement for manufacturing information which is specific to their own task: IP requires data on inspection probes; MP requires data on the machine tool, the cutters, the fixtures and machining rules; MDA requires data on faults and their causes.

Once data requirements for Applications had been identified, a further problem was to decide what data should be stored in the Product Model and what should be generated by Applications and discarded after use. At first sight it might seem sensible to store all product related data. However, this approach would lead to massive data storage problems as more and more data accumulated. The view was taken in the ISS research that data which could be readily, and consistently, generated should not be stored in the Product Model but generated as required; while data which may be manipulated and refined by a number of Applications must be

identifiable via the Product Model. An example of the former instance is seen in the use of the SDSM cell decomposition Application. This generates a large amount of data but, as it can be readily and consistently generated from existing product data, there is no need to store its resulting data within the Product Model. An example of the latter case is planning data. This data is required by, and could be modified by, a number of Applications. Hence a data structure should be used to enable its modification. This is different from the approach taken in the PART system [140], where the Product Model consists only of the initial design data.

The next stage of the research group's exploration of integration was to construct a data model within which the required product data could be represented. The next section describes the structure of the Product Data Model, constructed in the research programme, and provides a general background of the structure, highlighting, in particular, the data representations resulting from the author's research. These data representations are discussed in greater detail in the following chapter.

## **5.4 THE ISS PRODUCT DATA MODEL.**

### **5.4.1 The Framework for a Product Data Model**

Figure 5.2 illustrates the framework of the Product Data Model, within which the particular data representation issues addressed by the author were conducted. This framework introduces an initial level of structure into a Product Data Model, identifying key aspects of product data having greater or lesser significance to an Application, dependent upon the particular area of interest being addressed. A discussion of the framework, and the use of the repeating patterns described below, in product modelling, is given by McKay [94]. The framework has four levels of



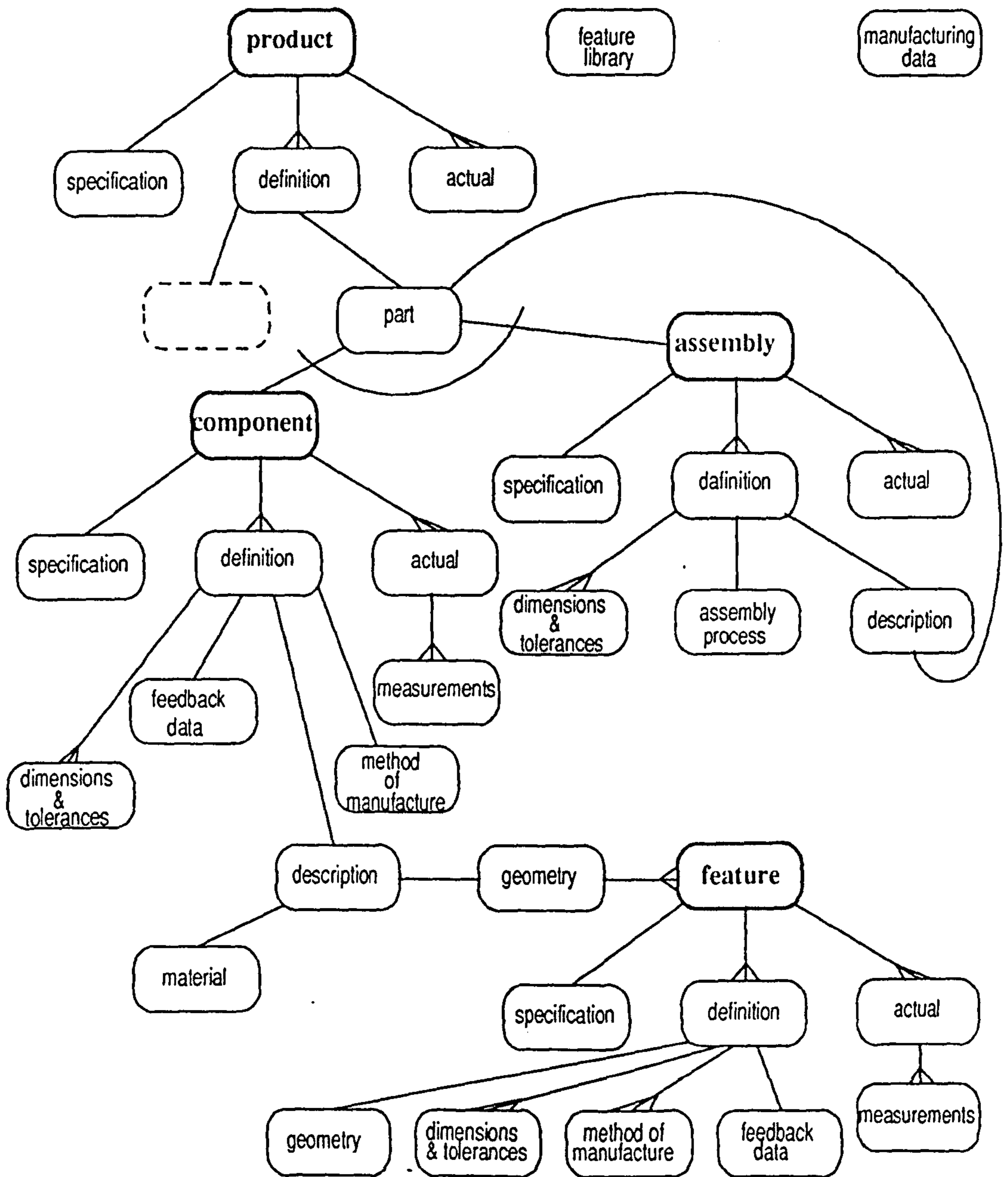
depth; highlighting product, assembly, component and feature entities. Within each level there is a repeating pattern of data representation, to describe the different types of data which are associated with entities. These are *specification, definition and actuals* where:

- *Specification*: describes the requirements which the actual entity must meet.
- *Definitions*: describe the ways in which the specification may be met. This will include designs and process plans for product entities.
- *Actuals*: describe data about actual entities which are produced e.g. measurements of inspected entities can be read from here and used in manufacturing data analysis.

Within the framework, attributes of entities were identified in the ISS project, which would provide the key elements of data of significance to MCG. These attributes can be described as follows:

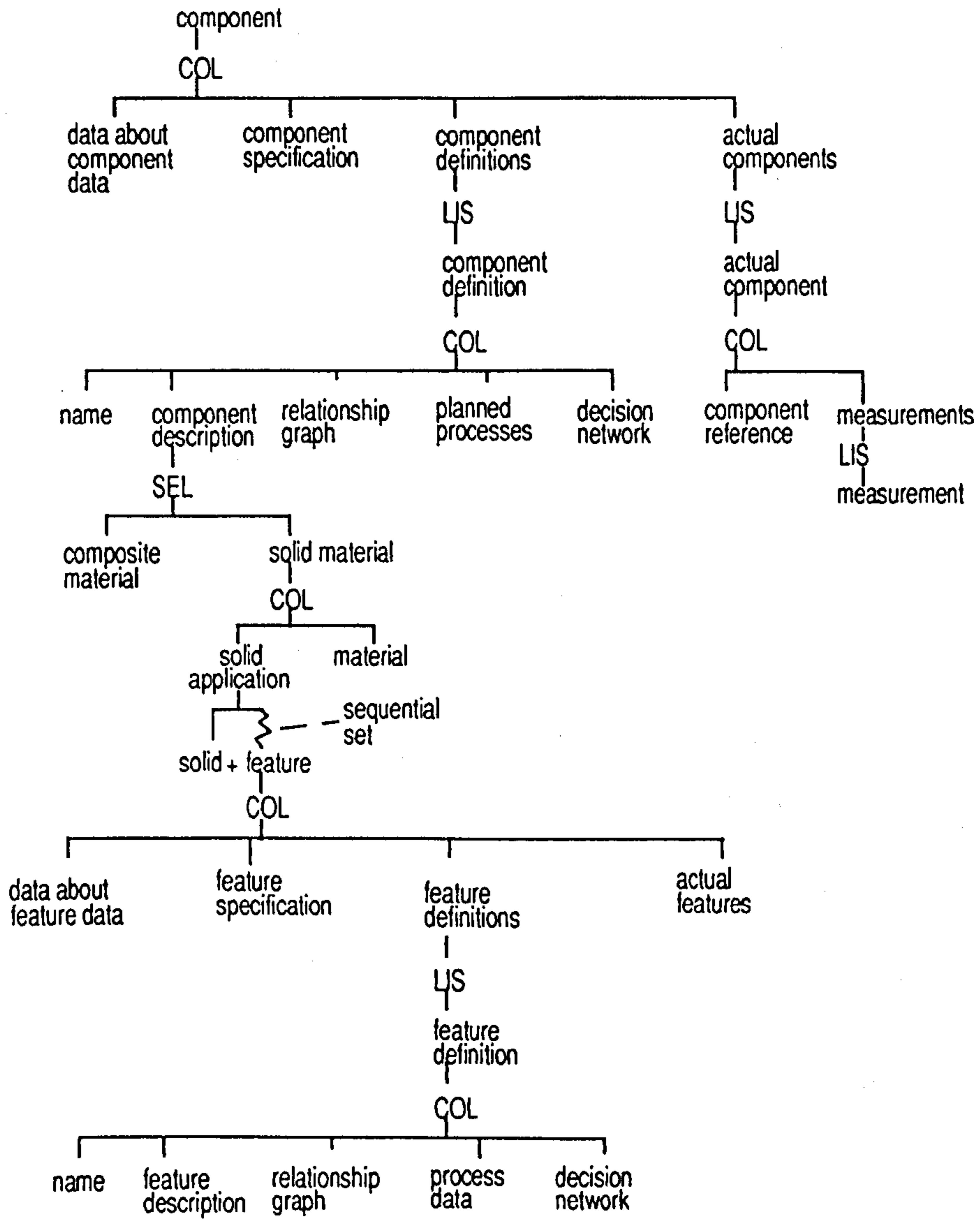
- The *entity description* which will depend on the entity, for example a component entity would be described here in terms of its geometry, features and material while a feature entity is described in terms of its geometry. Geometry in the ISS has been based on a constructive solid geometry (CSG) model which is itself based on the use of half space primitives. The definition of planning data to be associated with component and feature entities is discussed in chapter 6, while a method for the derivation of feature position and geometric interaction data from a component entity for Machine Planning is argued in chapter 7.





**The Framework of the Product Data Model.**

**Figure 5.2**



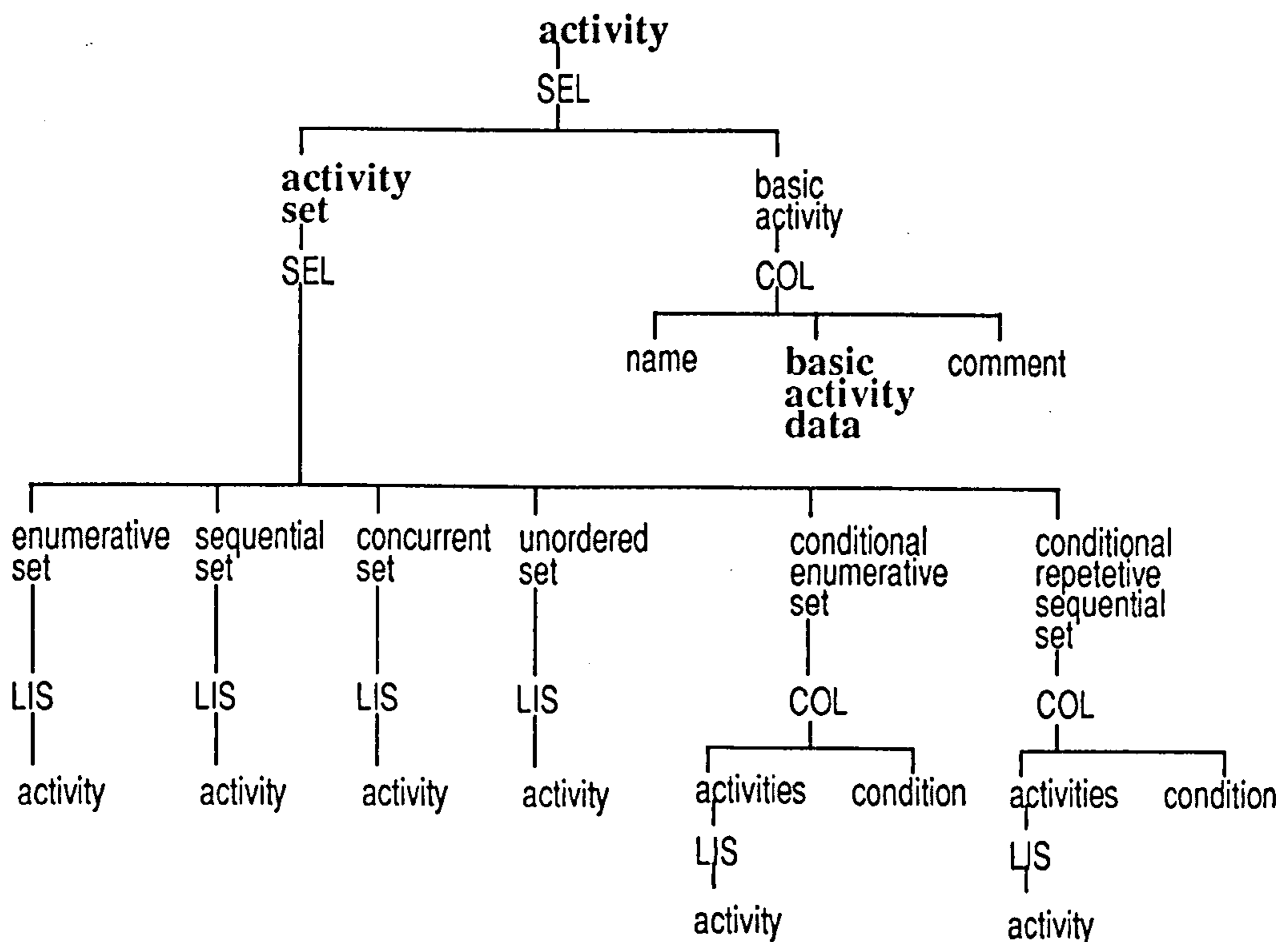
**The ISS Product Data Model for a Component  
Figure 5.3**

- The *dimensions and tolerances* which provide data in relationship to the geometry. This information is provided in the ISS programme by the *Relationship Graph* which is a syntax for dimensions and tolerances describing relationships between "dimension nodes". These dimension nodes can be linked to geometric faces, where dimensions are related to faces. Other dimension nodes, such as centre lines, can also be related back to geometric faces. Tolerances can also be linked to parametrised feature definitions. Links from dimensional representations to a Machine Planner are discussed in chapter 8.
- The *method of manufacture* which provides the planning data for the entity. The representation of data under this heading forms a key part of the author's research exploration. The data representation has been explored for feature entities as *process data* and component entities as *planned processes*. These aspects of a Product Data Model are introduced in subsequent sections of this chapter, with a detailed description of them being provided in chapter 6.
- The *feedback data* which provides the basis on which errors in manufacturing are analysed for error correction. This is represented in the ISS programme by a *decision network* which was defined and used in the Manufacturing Data Analysis work.

Other elements could readily be added to this set as additional, dissimilar, data requirements are identified for other Applications. With the above understanding of data requirements a component level data model was constructed, as illustrated in figure 5.3. This formed the basis for the integration of Machine Planning, Inspection Planning and Manufacturing Data Analysis, as illustrated in figure 4.5. The Machine Planning contribution to product data representation concentrated principally on the *process data* to be associated with features and the *planned processes*



associated with components, as highlighted in figure 5.3.



## Modelling Choices in a Data Structure

Figure 5.4

Choices in activity relationships are also important, with ISO STEP [109] identifying an activity structure as shown in figure 5.4. Their work identifies a planning activity as being either a basic activity or a set of basic activities. This structure has been used to allow for choices within the research. Activity sets can be a choice from: *an enumerative set*, which is a set containing a specific number of activities; *a sequential set*, which is an ordered set of activities; *a concurrent set*, which is a set of activities which can be performed at the same time as each other; *an unordered set*, which is a set with no order imposed upon it; *a conditional enumerative set*, which is a set where the number of activities depend upon some condition; *a conditional repetitive sequential set*, which is a sequential set which is repeated dependent upon some condition. The author's work has concentrated principally on

the use of sequential sets.

#### **5.4.2 Process Data Associated with Features.**

As described earlier, the significance of a feature can vary in its meaning for different Applications, as is the case with MP and MDA. Hence the feature data structure has been formulated such that a number of aspects of significant data can be associated with it. The structure defined in the ISS research has geometry, a relationship graph, process data and a decision network associated with it as is illustrated in figure 5.3. The *process data* is a key aspect of data for Machine Planning. The definition of the *process data* structure is explained in chapter 6.

#### **5.4.3 Planned Processes Data Associated with Components.**

The author's assessment of setup, operation and NC planning data requirements, using the project workpiece and pre-defined feature data, in conjunction with discussion with P. Bell and A. McKay on data requirements for automatic NC code generation, resulted in a manufacturing planning data structure. This *planned processes* structure provided the basis for the integration of setup, operation and NC planning, as well as a basis for interaction between pre-defined and recognised features. The author's contribution to the definition of this structure is described in chapter 6. The work of Bell, McKay and the author, on the definition of this structure, has been presented to the STEP committee [95].

## 5.5 THE USE OF THE PRODUCT DATA MODEL TO LAY THE FOUNDATION FOR INTEGRATION.

With the previously described Product Data Model, integration between the Manufacturing Code Generation Applications can then be achieved by (i) using *definition* data in the form of geometry, features, dimensions and tolerances and material to produce *definition* data in the form of machining plans in the planned processes structure; (ii) using *definition* data in the form of geometry and dimensions and tolerances to produce *actuals* data in the form of measurements of the inspected component. (iii) using *definition* data in the form of dimensions and tolerances, features, material and planned processes as well as *actuals* data in the form of measurements to analyse the component and feed back any errors. This is illustrated in figure 4.5.

The capability of the Product Data Model, described in the previous section, to provide an integrating mechanism for Manufacturing Code Generation was explored by representing a particular component in the framework and using this to provide data to the MCG activities. This experiment is described in chapter 11.4.

The result of the above research was to show that Manufacturing Code Generation Applications can be integrated through the use of a common base of product data, and that the ISS Product Data Model provides an appropriate structure for this data. Hence future design and manufacture systems can be anticipated which draw on the benefits which can be achieved by enabling the interaction of Applications through an integrated system. The detailed aspects of the integration research particular to Machine Planning, and contributing to the overall achievement of integration, are described in the following chapters.



## CHAPTER 6

### MACHINE PLANNING - A CLOSELY COUPLED APPLICATION.

#### 6.1 INTRODUCTION.

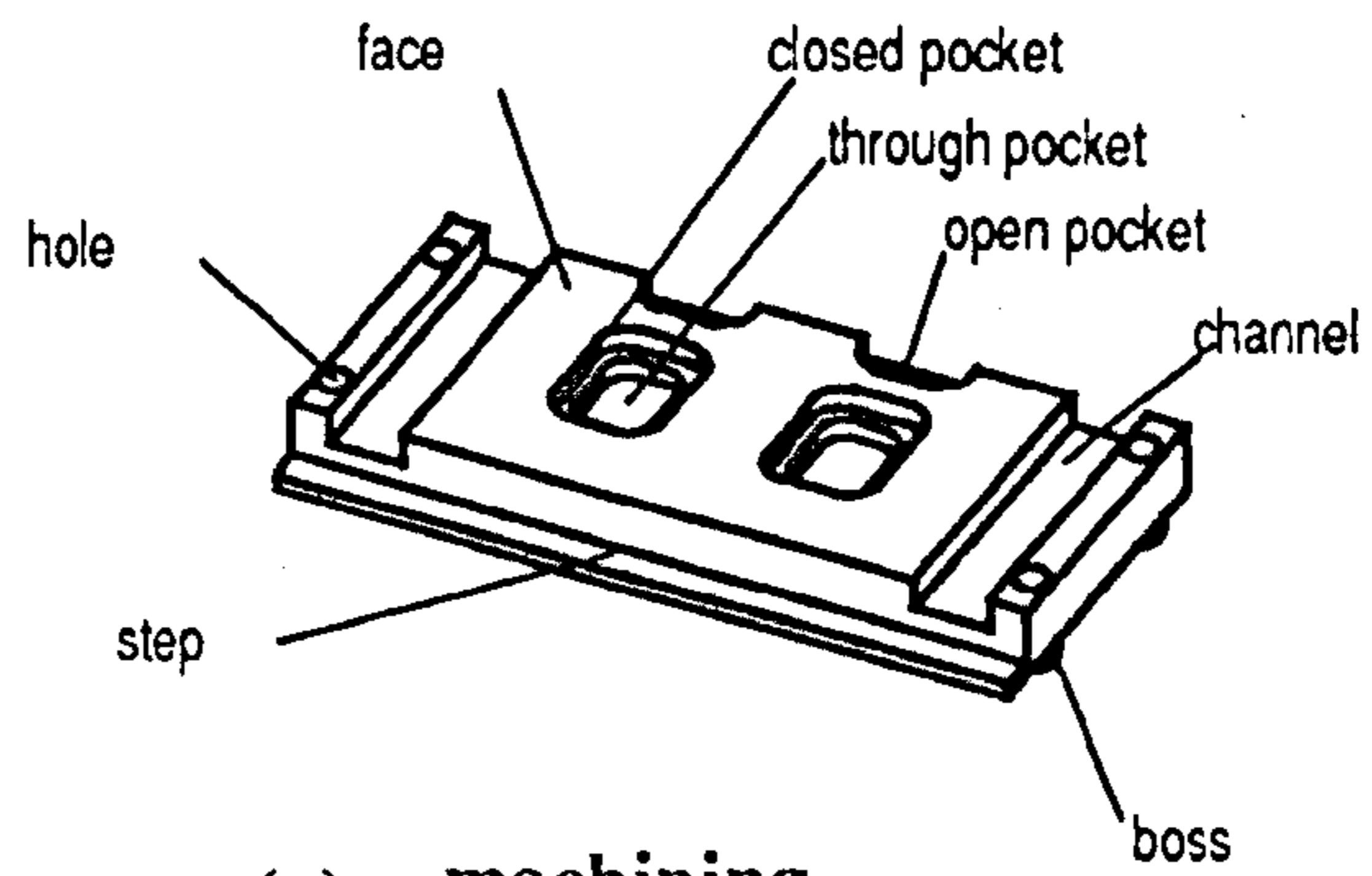
This chapter describes the research performed by the author to define the nature of the lower levels of a Product Data Model which can provide the basis for the integration of Machine Planning activities. Product data constraints on the planning of the machining of discrete features and interacting features have been considered. A structure for Machine Planning activities and their interrelationships is also discussed. This has resulted in the derivation of the feature level *process data* sub-set of the Product Data Model and the component level *planned processes* sub-set of the Product Data Model, described in chapter 5, as well as a view of how these can be used in the integration of Machine Planning.

The Product Data Model sub-sets argued in this chapter have evolved over a period of time, during which earlier versions of a PDM were constructed and tested, as described in chapter 10.6. For clarity, only the final PDM, which encapsulates the requirements identified from the initial explorations, is defined here. The scope of the experimental Machine Planner, used to explore the PDM's ability to provide an integration capability, is explained in chapter 9, while the way in which the sub-sets of the Product Data Model have been used in the realisation of an experimental Closely Coupled Machine Planner is described in chapter 10.

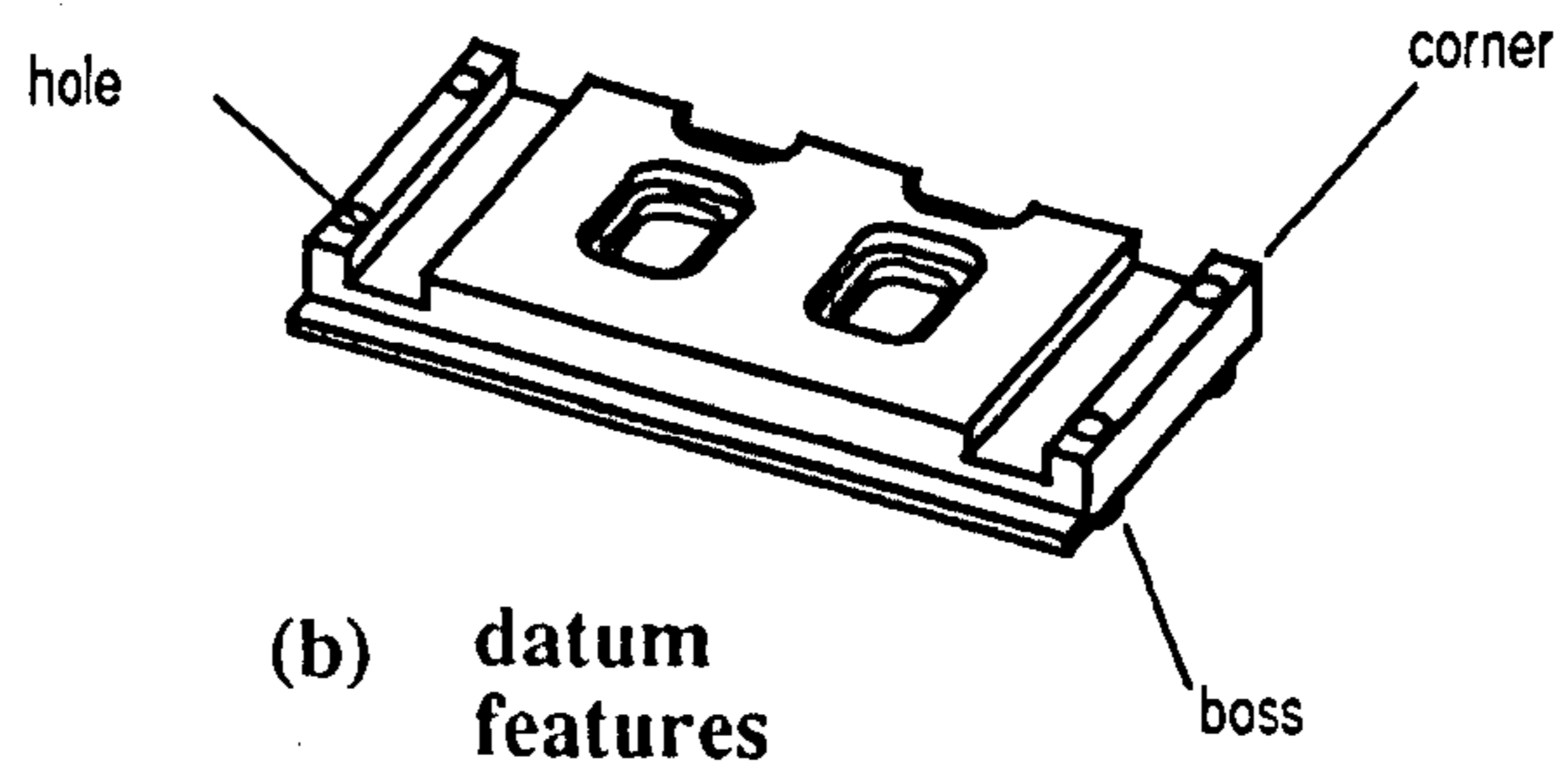
## 6.2 THE ROLE OF FEATURE DATA IN A MACHINE PLANNING APPLICATION.

This section describes the general view taken of features in this research, and how they influence the Close Coupling of a Machine Planner. It has been understood for some years now [20] that features play a key role in providing integration links between design and manufacture, with much research effort being channelled towards methods for the recognition of features from geometric descriptions, as described in the literature survey. The STEP organisation have addressed a structure for features [108] as part of their work to provide a data integration standard. Their work has also concentrated on the need to provide a geometric understanding of features. The emphasis of the author's work has been on process data structures related to features and components, exploring a view of feature based machining from which to derive the lower levels of a Product Data model which can, in turn, provide a basis for Machine Planning integration.

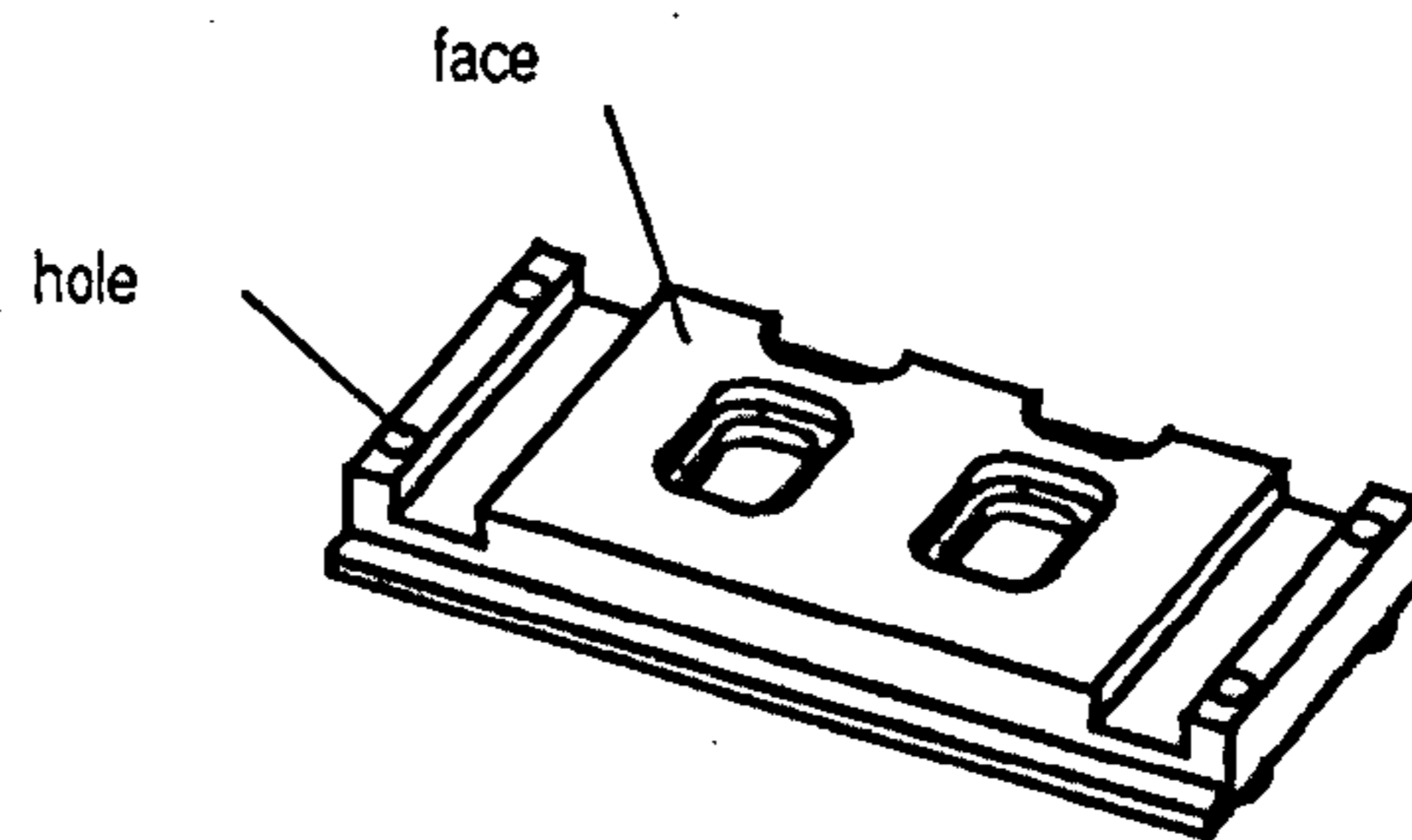
The main function of a Machine Planner is to plan the way in which cutters should be used to machine a workpiece. Fixturing and datuming the workpiece are necessary functions required to meet the main aim. To machine, fixture and datum a workpiece requires an understanding of the significance of geometric shapes for these three purposes. Therefore three types of feature have been considered important in this work; machining, fixture and datum features. Examples of these are illustrated in figure 6.1 which uses the Bolster Plate benchmark workpiece. This workpiece provided a focus for the work of the members of the ISS project as explained in chapter 5 and versions of this, using varying levels of complexity, are used throughout this thesis.



(a) machining features



(b) datum features



(c) fixture features

**Examples of Feature Types  
in Machine Planning**

**Figure 6.1**

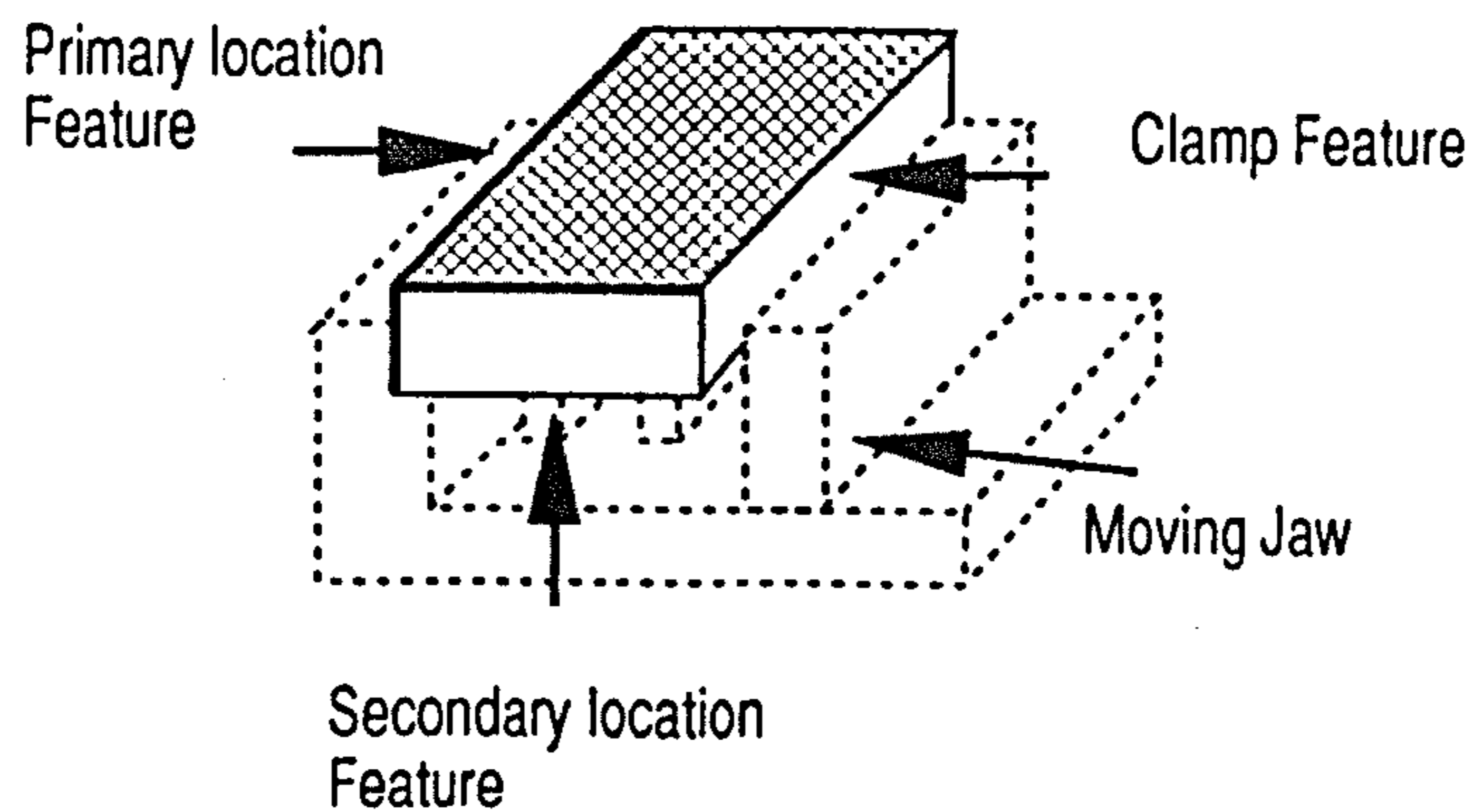


### 6.2.1 Datum and Fixture Features.

Datum features provide regions of the component from which the position of the component on the machine table can be defined. Three datum feature types have been identified. These are holes, corners, and bosses as illustrated in figure 6.1. Each of these provides a set of related faces, whose position with respect to the x, y and z axes can be found which in turn identifies the workpiece position with respect to the machine table. Datums can be located on the workpiece, or on a fixture. In this work they have been included in the definition of fixturing strategies as described in chapter 9.2.3. How datum positions on machined or un-machined surfaces have been accommodated is described in chapter 7.3.1.

Fixture features provide regions of a workpiece which can be used for fixturing. Three types of fixture feature have been distinguished as being required for workholding and are defined as follows: Firstly, Clamp features are to be used to clamp the workpiece. Secondly, Primary Location features provide the main location surface of the workpiece against the fixture, a surface opposite to the clamping direction. Lastly, Secondary Location features provide further location to ensure the workpiece is held rigidly in position. The illustration of a rectangular block in a vice in figure 6.2 provides an example of such clamping and location features.

In each case above, the geometry required to fulfil fixturing requirements are either faces against which clamps can be used, or holes, through which bolts can be used. Through holes and the external faces of a workpiece are obvious candidates for fixturing and their use has been investigated in this work. These feature types can be identified readily from feature names, as explained in chapter 9.3.1, and used in the identification of appropriate fixture features, as explained in chapter 9.3.2. The identification of the suitability of other faces on a workpiece for fixturing could be

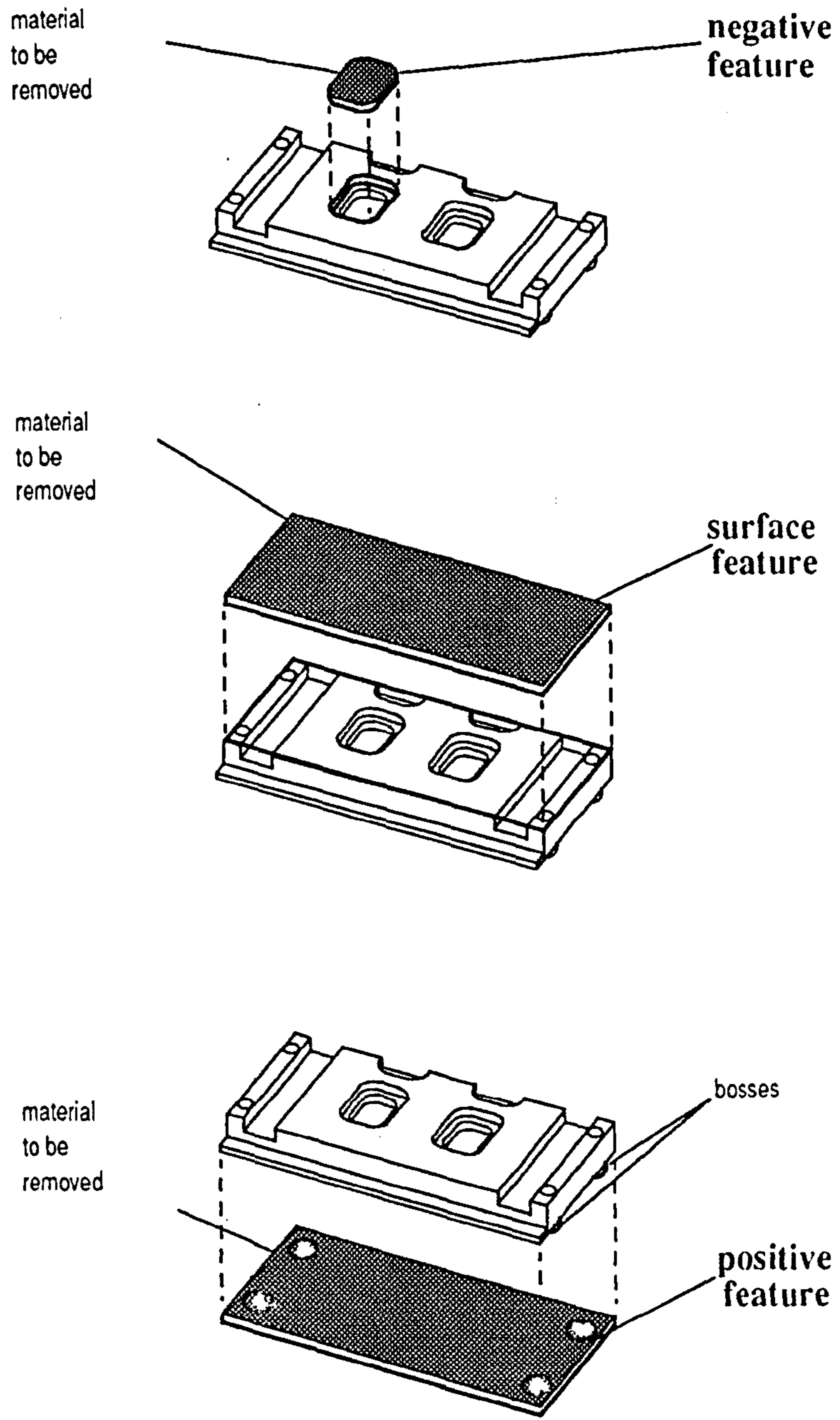


**Types of Fixture Feature**  
**Figure 6.2**

explored through the analysis of the component geometry. However, this would involve a level of complexity of feature recognition which would distract from the main emphasis of this research and hence no other fixture feature has been included in the investigation.

### 6.2.2 Machining Features

Research into the identification of machining features on a workpiece has generally concentrated on the identification of geometry which can be related to specific machining methods. For example, Choi [30], described commonly defined machining feature geometries as holes, pocket, slot, step, 2D-contour, plane, and sculptured surface. His work then explored methods of recognising such geometric shapes, as has most subsequent research on machining features. Different classifications of feature have also been used to clarify the geometric significance of features. Typically features are classified into depressions and protrusions [24], or depressions (negative), surfaces, and protrusions (positive) [52]. An example feature from each category from the latter classification is illustrated in figure 6.3. As can be seen from the figure, the importance of each feature type is the material which needs to be removed to produce the feature, and not the feature itself.



**Machining Feature Types**

**Figure 6.3**



The importance of machining features in this research is to capture the structure of process data, related to the material to be removed from the feature, rather than to explore methods of feature recognition. The structure of process data gives a feature meaning, such that a Closely Coupled Machine Planner can extract, and use, the data from a Product Model.

In section 6.4 a structure for a feature level process data sub-set of a PDM is argued, which has been used in the realisation of the Closely Coupled Machine Planning Application, as described in chapter 10, and used in the experimental work discussed in chapter 11.

### 6.2.3 Feature Interactions.

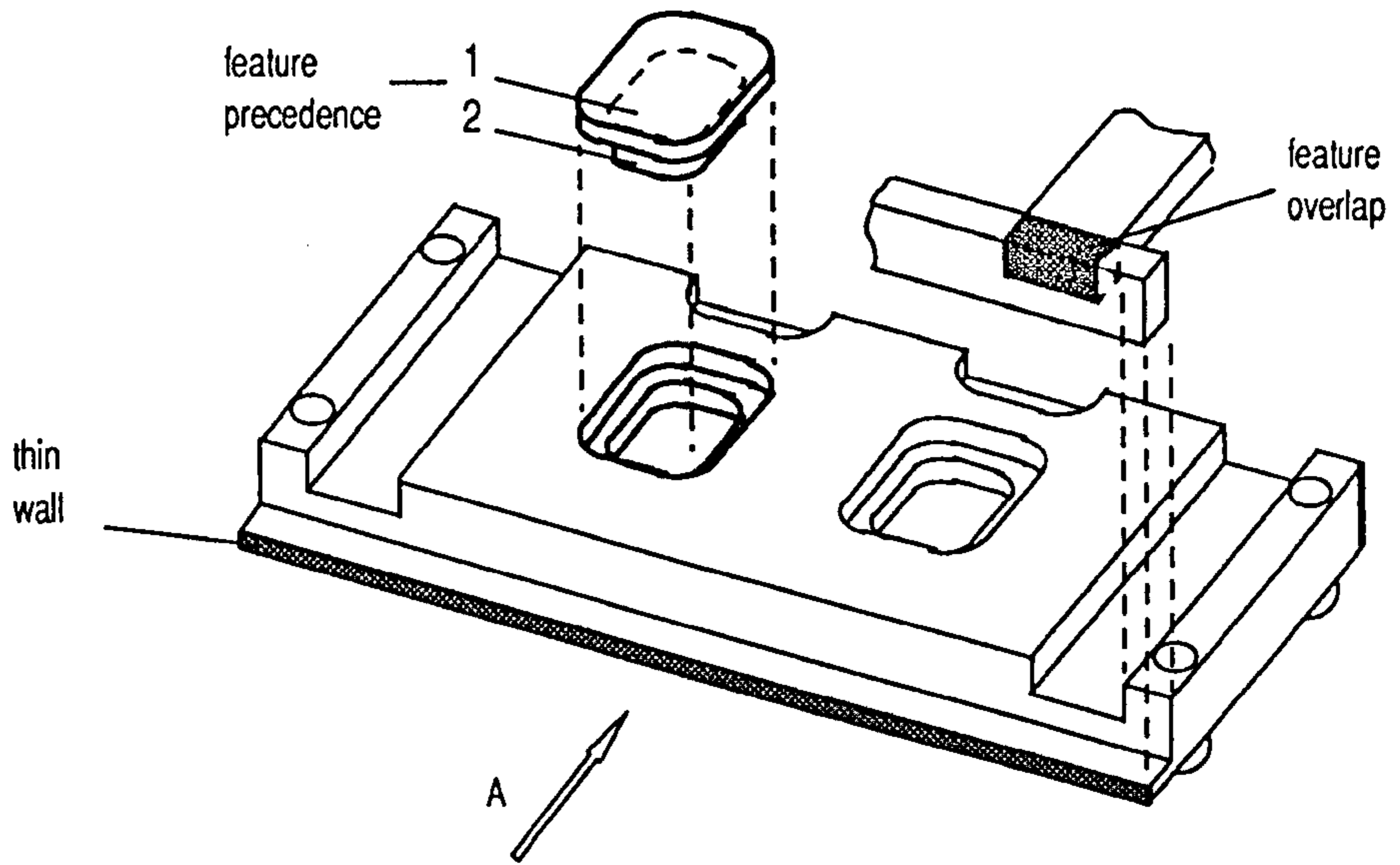
Feature interactions are important to the author's research where they influence the product data employed in a Closely Coupled Machine Planner. The interactions between features can be thought of as providing geometric or tolerance constraints on the planning methods which can be employed in producing a workpiece. These are discussed in the following paragraphs. The results of interactions between features are not directly related to the structure of specific features themselves, but relate to the planning methods to be used in machining the component as a whole. As such, the component level *planned processes* is needed to capture the progressive data generated through setup, operation and NC planning.

The interaction of machining features within a setup can introduce three possible geometric constraints which may influence machining. These are feature precedence, thin walls and feature overlaps which are illustrated in examples shown in figure 6.4(a). Feature precedence occurs when a feature sits on top of another feature and therefore must be machined first. An example of this, in figure 6.4(a), is the "pocket 1" which lies over "pocket 2". Feature precedence was used by

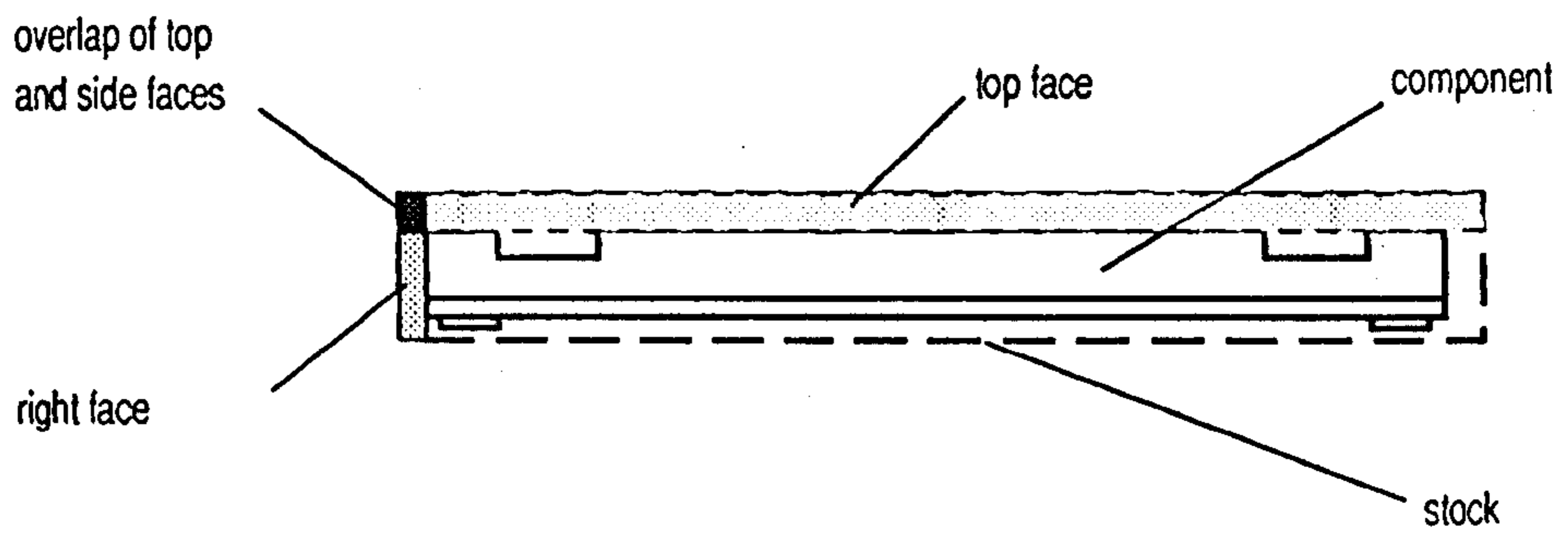
Joshi [73], as a means of sequencing the machining of features. Although this must influence an operation sequence, other technological factors must also be considered as described in section 6.3.2 of this chapter. Within the ISS program, feature precedence routines were pursued by Bell [12], who explored a Region Graph to define the relationships between features in a setup.

Thin walls between features occur when machining features are applied close to each other, such that the wall between them is likely to be distorted by normal machining forces. The need to identify such feature interactions was identified by Willis [149]. How thin walls between features can be identified, by linking to the SDSM, is described in chapter 7.4.4.

Feature overlaps can lead to the initial processing method of each feature becoming void. For example, if two pockets are merged together it would be appropriate to reassess their machining as a single pocket. This however, introduces a need to explore feature recognition to identify how such updates can be made. The structure of data associated with the feature will not be changed by this, although the data content will be different. Therefore the problem of overlapping features which negate the defined processing method has not been pursued in this research. Other feature interactions have been addressed.



(a) Examples of Overlap, Precedence and Thin Wall Interactions

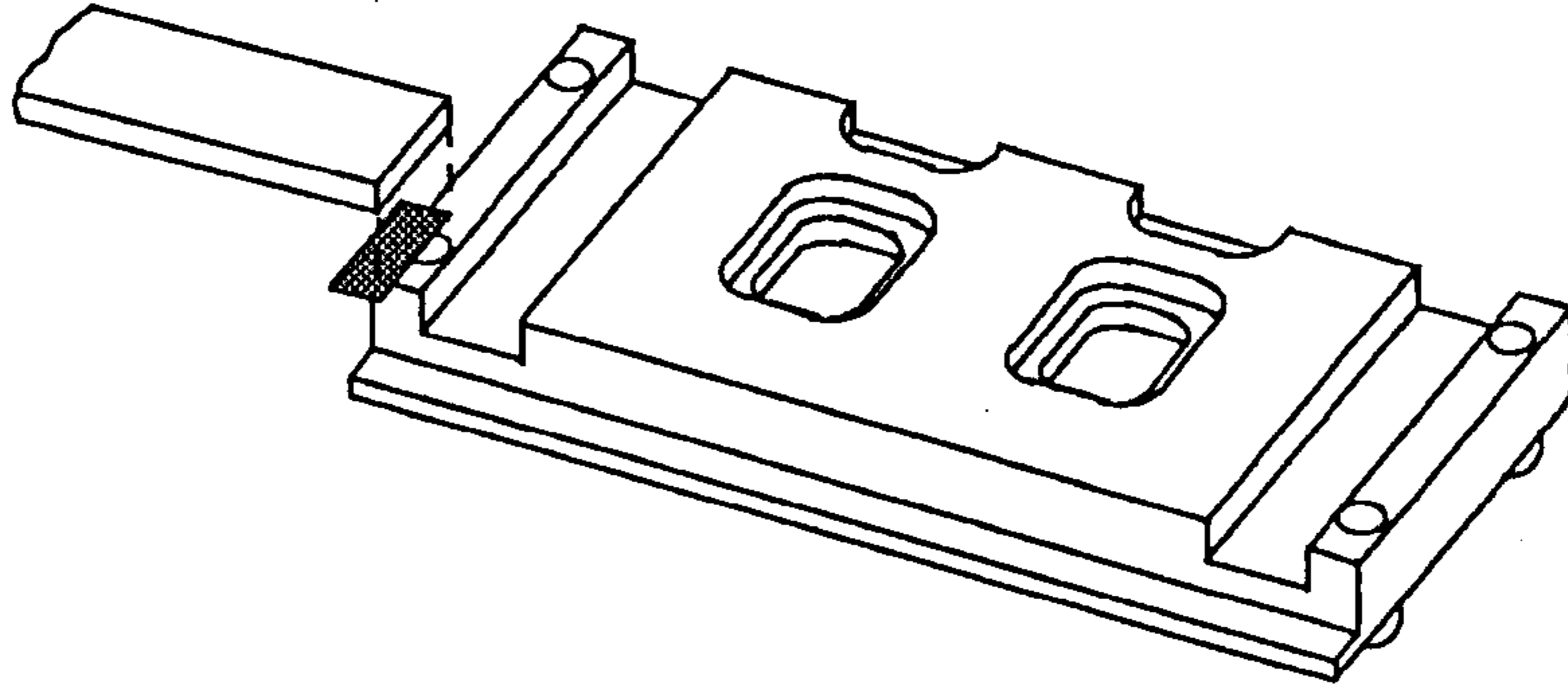


(b) View in direction of arrow 'A' showing stock around component

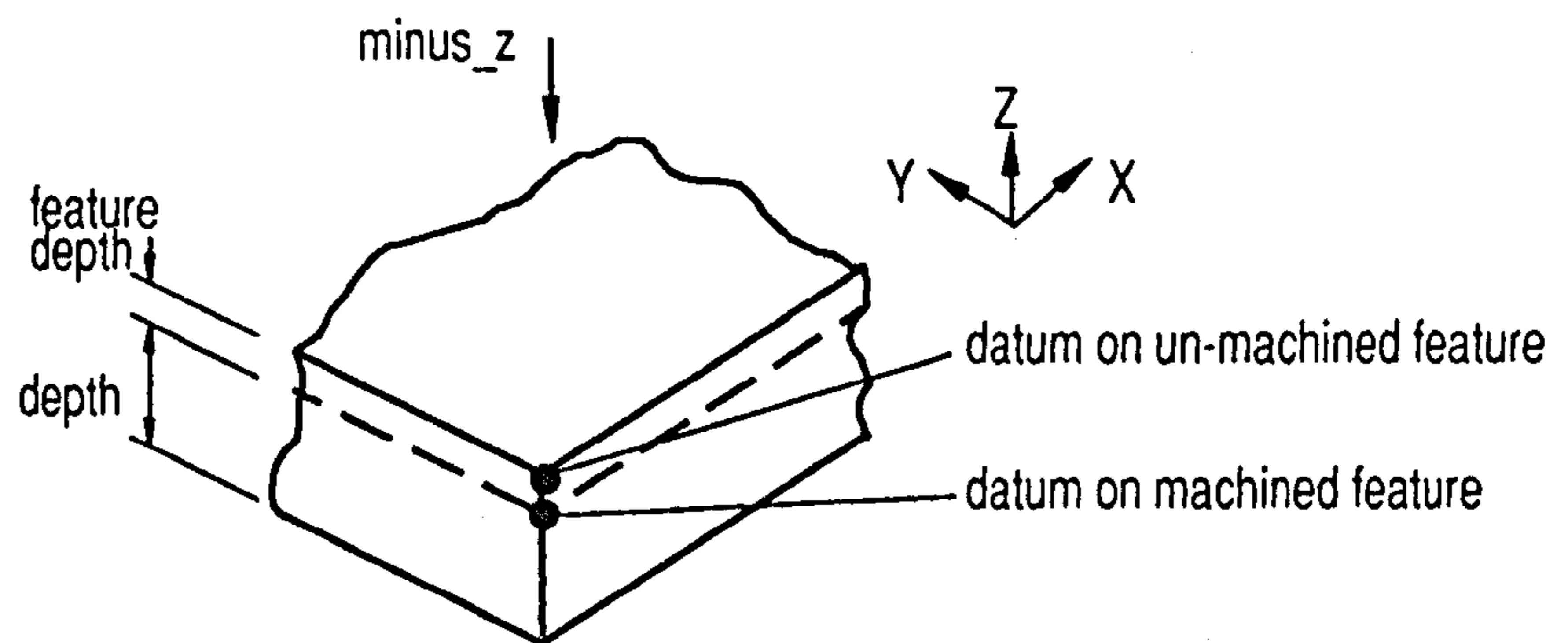
**Geometric Interactions  
between Machining Features**

**Figure 6.4**





(a) Fixture and Machining Feature Interaction



(b) Datum and Machining Feature Interaction

## Fixture and Datum Interactions with Machining Features

Figure 6.5

The geometric interaction of machining features across multiple setups introduces two further interactions to be considered: the stock relative to the component and the prospect of partial machining of a feature in one setup. Figure 6.4(b) illustrates these. The stock relative to the component must be defined such that the surfaces of the workpiece will clean up to achieve the component dimensions. This positioning will be influenced by the parameters to be associated with Face features on the component. How this requirement has been accommodated in the software implementation is described in chapter 10.4.3 and 10.4.4. Face features are also likely to be partially machined in some setups as illustrated in figure 6.4(b). This results in the need for geometric updates as described in chapter 7.3.4.

In summary the geometric constraints introduced by interacting machining features do not change the feature *process data* structure but they introduce the need for geometric analysis techniques to identify interaction problems. The use of the SDSM to provide a basis for such analysis is described in chapter 7.4.

Machining features may also interact with fixtures and with datums as illustrated in figure 6.5. The former interaction problem has been identified by Kanumury [76]. When fixtures are positioned on a workpiece it is necessary to check that the fixture will not overlap any machining features which are to be produced in the setup. How the SDSM has been used to provide queries to check such problems is argued in chapter 7.4, while the resulting software implementation is described in chapter 9.

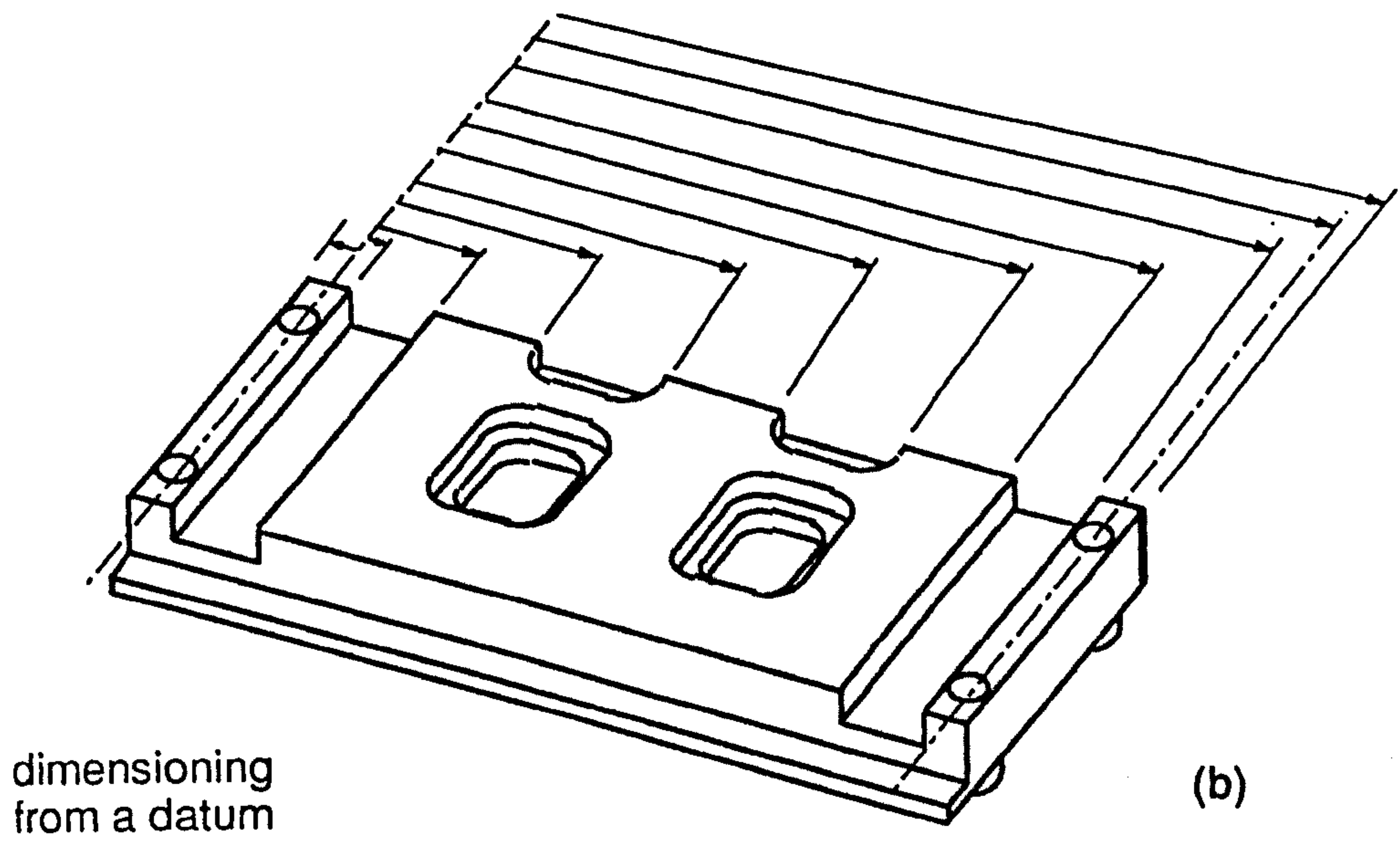
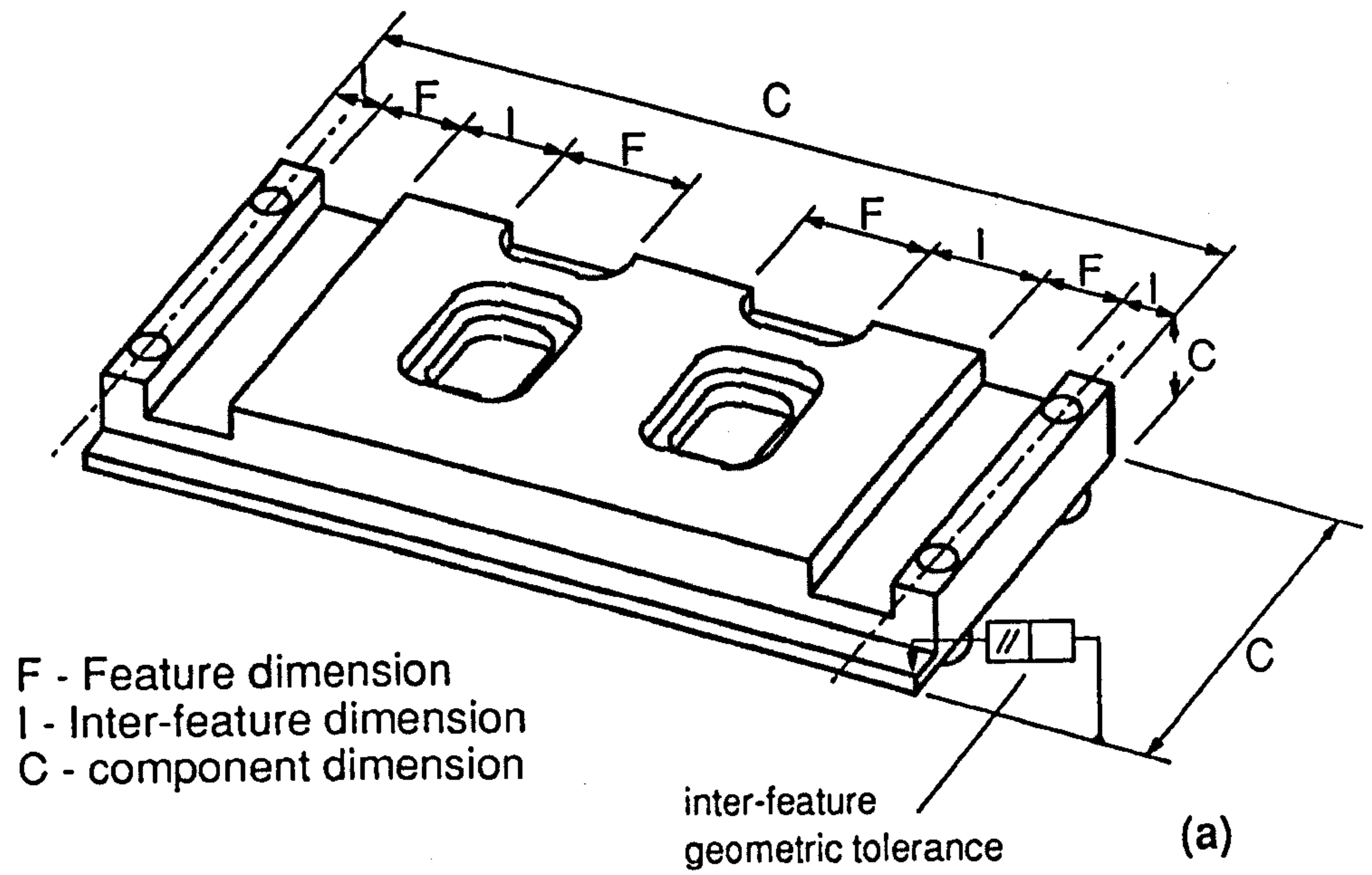
The effect of machining features on datum positions is to move the datum position relative to the component, dependent on whether the machining feature has been produced before or after the datum has been set. How datum positions relative to the component have been checked is described in chapter 7.3.1

Features also interact through tolerance constraints. Dimensional and geometric tolerances, and surface texture, influence the manufacturing methods which can be used on components. The tolerances which can be achieved when machining will be dependent on the repeatability of the position of the machine tool spindle, the accuracy of the measurement of the cutter, the accuracy of the machining performed, and the accuracy of location of the workpiece with respect to the machine tool spindle from setup to setup.

Dimension and tolerance constraints on components can be represented in different ways. Two options are illustrated in figure 6.6, where it is assumed that all dimensions will carry some tolerance value. The first uses feature and component based dimensions. These can readily be represented in a feature based component model. However, as can be seen from figure 6.6(a), there is still a need for inter-feature dimensions to be represented, both for dimensional and geometric tolerances.

Inter-feature dimensions, by their nature cannot be stored within the feature data sub-set of the Product Data Model. However dimension and tolerance relationships between geometric faces can be associated with a component as a Relationship Graph [148]. How this representation of component dimensional relationships could be analysed, in conjunction with an understanding of machining features, to derive inter-feature tolerance data is discussed in chapter 8. The re-evaluation of a dimension and tolerance scheme to generate alternative schemes such as shown in figure 6.6(b) is also discussed.





**Dimensional Interactions  
between Machining Features**

**Figure 6.6**

#### 6.2.4 A Comparison of Machining Feature Recognition and Pre-definition.

To explore the representation of a process data structure related to features, the author has had to select a means by which this exploration could take place. This could either be through the use of pre-defined features or through feature recognition. This section compares these two approaches and explains the author's choice of using the pre-defined feature route, while exploring a data structure which can offer the prospect of integrating the benefits of both approaches.

The concept of feature pre-definition is that known features should be defined as parametric descriptions, stored in a feature library, and then used in the process of component description. The advantage of this approach is that prior knowledge of manufacturing methods can be stored with features and extracted from the component definition as required. No extensive analysis of the component description is required to identify the data. Designers can also be encouraged to design using known features to achieve a level of standardisation in approach.

A disadvantage of this approach is that designs are effectively restricted to the use of pre-defined features which may overconstrain the designer. Only simple shapes, which are used regularly, are worth storing in a library as complex shapes become difficult to define parametrically. Since features are defined as individual entities, only data relating directly to them can be stored. Data relating to interaction effects with other features must be found by some other means.

Although these restrictions on the value of the pre-defined features approach are recognised, the overriding value to the author's research is that pre-defined features provided a medium through which the data structures identified in this research could be investigated. They also allowed feature interaction problems to be



analysed assuming the process data associated with each feature was not invalidated by that interaction. Hence, they provide an appropriate medium for the exploration of a Closely Coupled Machine Planner.

The concept of feature recognition is that a component description can be analysed to identify the features from which it is composed. This approach starts by analysis of an existing component description. As such no feature based restrictions are placed on how the component description is generated. Therefore this approach has the advantage of flexibility in component definition. Further flexibility can be achieved in the feature analysis as it can, in principle, be conducted from whichever viewpoint is appropriate. For example, alternative analyses could be conducted to identify machining, fixturing, datum or assembly features.

Feature recognition is not an easy task however. A great deal of research effort has been spent on assessing how recognition can be achieved and this work is still ongoing. The PART system [140] of feature recognition is perhaps one of the most advanced, providing a basis for analysis which can cope with feature interactions. Feature recognition, although offering greater flexibility in principal, than feature pre-definition, has not been considered essential to the main thrust of the author's research. Links to the geometric analysis based work of P.Bell [12] has however, been considered.

The author's work in conjunction with Bell addressed how Bell's geometric analysis approach could be integrated with the pre-defined feature approach used by the author. Activities MP11 and MP13 of the Machine Planning IDEF0 model in appendix 3 were produced to reflect how these two approaches could be linked. As a result of this work, the *planned processes* sub-set of the PDM, described in section 6.5, has been defined to include structures which can form the basis for an approach combining the benefits of both feature recognition and pre-definition.



### 6.3 MACHINE PLANNING ACTIVITIES AND THEIR INTERACTIONS.

Ham [57] identifies the need to have a clear understanding of the tasks involved in process planning and their relationships if progress is to be made in integrating and automating process planning functions. The need to employ modelling techniques to explore the definition of these relationships is being recognised in the international research community. The ESPRIT IMMPACT programme [68] is pursuing information modelling as the key to integration and automation of design and manufacture processes. CAM-I have been using IDEF0 as a route to defining process planning interactions as described in chapter 5.2.3. Chapter 5.2 describes the ISS project's use of IDEF0 to represent the activity relationships between the members of the project and breaks these down as far as Machine Planning.

Chan [25], in exploring a machining process language to aid process planning, also took the view that the structure of planning tasks is important. He introduced a view of the relationships between process, setup and operation planning which proposes setup planning as a main task in process planning, with operation planning being a downstream task. This is not the view taken by the author as setup planning is dependent on cutter accessibility and the identification of appropriate cutters is seen by the author as an operation planning problem. The author's view is that setup and operation planning are interlinked; that operation planning and NC code generation should be interlinked and that these three planning tasks are related to Machine Planning which then can feed back to process planning as required.

The following sub-sections address planning for the machining of discrete operations, then features in a setup and then features across a number of setups. This progressive assessment of planning more complex machining situations resulted in the Machine Planning activity model described in appendix 3. This in turn has pro-

vided the scope for the Machine Planner implementation described in chapter 9, which has enabled the research exploration to be performed.

### 6.3.1 Planning the Machining of Discrete Operations.

An operation is defined here to be a volume which is to be machined by one cutter. When machining an operation decisions have to be made as to which cutter to use, with which cutting parameters and with which tool path. Selecting cutting parameters and tooling paths can be considered as part of NC planning, while selecting which tool to use forms a part of operation planning. Normally NC planning would be considered to be a subsequent activity to operation planning. However, selecting the most appropriate tool to machine an operation is partly dependent on the machining time, which in operation planning is usually estimated. Integrating NC planning with operation planning therefore provides the advantage that path times can be accurately simulated, resulting in more accurate planning.

Path times for a single operation can be considered as the sum of the time taken for the cutter to reach the start point for cutting plus the time from the start point for cutting to the end of machining the operation. The time taken to reach the start of the operation depends on whether the tool being considered is in the machine tool spindle or in the machine tool magazine. The time taken from the start to the end of machining the operation will depend on the cutting parameters used and the tool path used. Hence data on the machine tool and cutting tools available is required in addition to product related data when planning the machining of discrete shapes. The influence of tool paths and cutting data selection on tool selection has recently been recognised by CAM-I, who are now proposing to explore this area of work through their links with TNO/CRIF [113].



A wide range of cutting tools are available for use on machining centres such as drills, reamers, boring bars, face mills, end mills, slot drills, disc cutters for slotting and side milling, ball nose cutters, as well as non standard tooling prepared for particular machining requirements. Tool selection for hole machining [92,26] has been comprehensively researched, while tool selection for more general milling has received less attention. Melkote [96], has considered the selection of milling cutters to identify which tool in a catalogue is best for a specific operation. Face mills and end mills were considered in his paper and assessed against tooling characteristics such as rake angle, lead angle and diameter. In his work the user provides the input of the type of machining being performed and he identifies a requirement for process descriptions of operations to be provided from the component geometry.

It has been recognised [24] that specific tools should not be chosen for an operation without regard to the other operations to be performed during the machining sequence. These other operations can influence the appropriate selection of cutters by reducing tool change time and as well as the number of cutters required in the machine's carousel. Therefore the approach taken in this research has been to identify *possible* tools which can be used in machining specific operations through a tool selection procedure. Thereafter the *chosen* tool to be used can be identified during operation sequencing which must be performed when machining a set of operations in a setup. The software functions implemented to identify cutting tools and calculate processing data are described in chapter 9.3.3, and have been used in the experimental exploration of operation data structures, as described in chapter 11.



### 6.3.2 Planning the Machining of Interacting Features within a Setup

As described in the previous section, the activities involved in the machining of discrete shapes are tool selection, cutting parameter selection and NC code generation. These activities also apply to interacting features within a setup. The difference here is that there is a choice in the order in which operations can be performed. The problem is therefore to decide on the sequence of operations, and the specific tools to be used for each, starting from a point where an unsequenced set of operations have been identified, each with a set of possible tools, cutting parameters and path time data. Producing a sequence involves exploring the technological and geometric constraints which apply to the operation set, ensuring that these are met while attempting to minimise the time taken to machine the setup.

Constraints which can apply to the sequence are:

- geometric precedence,
- tolerances constraints, restricting operations to the use of a common cutter,
- technological constraints, such as machine roughing operations before finishing operations

The application of such constraints to a set of operations will produce a sequence of sub-sets of operations. These then can be assessed against the possible operation and tool combinations to minimise the time taken to machine the setup. The software implementation of an operation sequencing procedure is explained in chapter 9.4, and its use, to aid the integration exploration, is described in chapter 11.

### 6.3.3 Planning the Machining of Interacting Features across Multiple Setups

When multiple setup interactions are considered, the way in which setups are planned must be addressed. Setup planning is concerned with determining the setups in which operations should be machined, identifying the work-holding methods and datums which should be used, and the sequence in which setups should be machined. The aim in setup planning is to minimise the cost of machining, whilst still achieving the required specification of the workpiece. This is usually achieved by minimising the number of setups required.

To gain some insight into the constraints which apply to setup planning, and how these influence the Close Coupling of a Machine Planner, discussions were held with industrial part programmers from Lucas and British Aerospace, as well as technicians within the Department of Manufacturing Engineering at Loughborough University of Technology. Following from these a number of factors which influence decisions on setup planning can be described as follows:

- the number of components in a batch, and whether different components should be produced at the same time.
- the type of stock e.g. cast, forged or solid,
- the type of machine being used,
- the available fixtures,
- suitable features of the workpiece for fixturing,
- the interaction of fixturing with machining,
- problems affecting dimensional accuracy,

More than one component may be fixtured to the machine table in any one setup. In this case it will be necessary to consider the different component setups and amalgamate them into machine table setups. As the ISS project was constrained to the problem of addressing integration for single components this problem was not addressed further, other than to register it as an area of work which will need to be addressed in the future.

The type of stock being used will influence how the workpiece can be held and, hence, the way in which setups can be achieved. This relates directly back to the need to identify fixture features which can be used for fixturing the workpiece as described in section 6.2.1.

The type of machine to be used also influences setup planning. Consider, for example, horizontal versus vertical machining centres. The orientation of the spindle axis to the machine table is different for both of these, which influences how the workpiece can be fixed to the table when a particular group of operations is to be machined. The addition of a fourth axis on a machine introduces the ability to rotate the workpiece with respect to the machine table, therefore enabling operations with different directions of access to be included in a single setup. Hence different machine tools offer the ability to use different fixturing strategies.

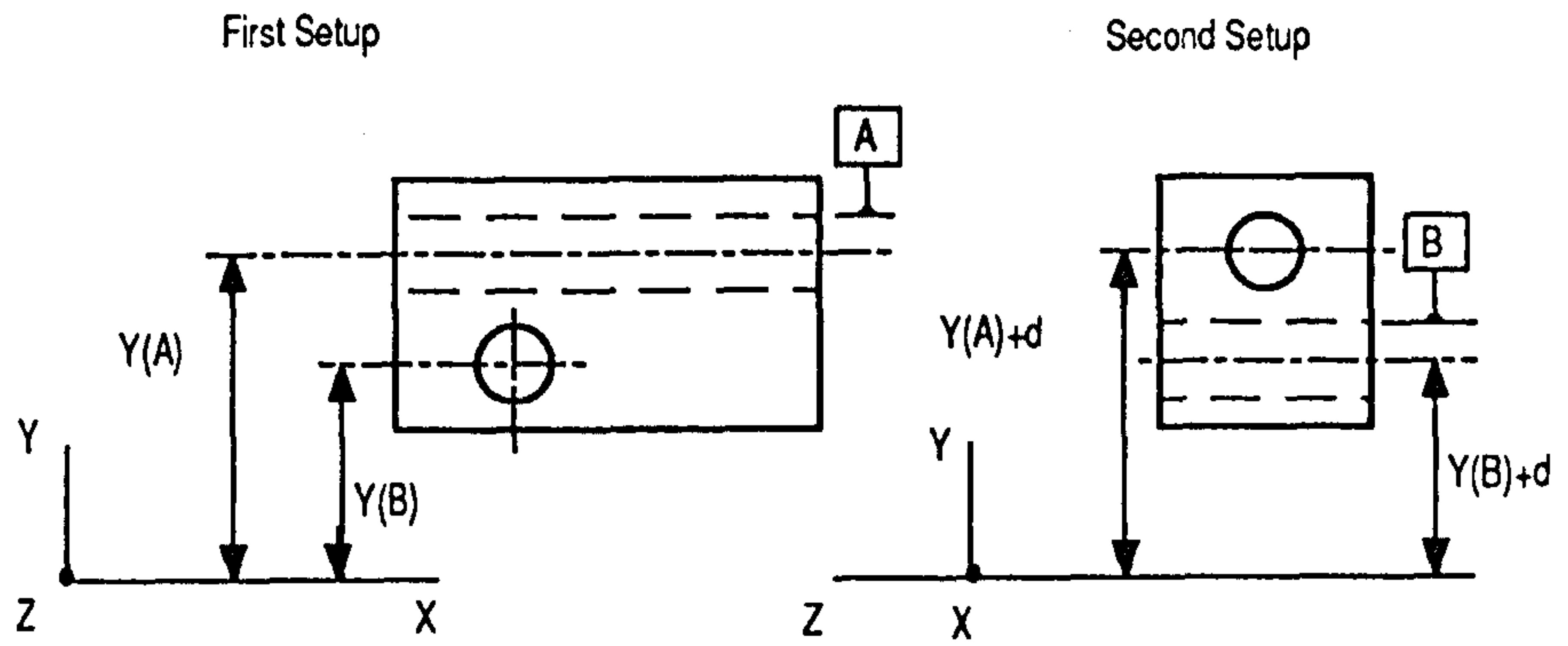
The selection of setups and design of fixtures, based on an analysis of the tolerance relations between features has been researched by Boerma and Kals [17,18]. They provide perhaps the most comprehensive research to date on tolerance relationships and their effect on setup planning, but do not address the problem of geometric interactions between fixtures and the workpiece as described in section 6.2.3 or the possibility of assessing different fixturing strategies.



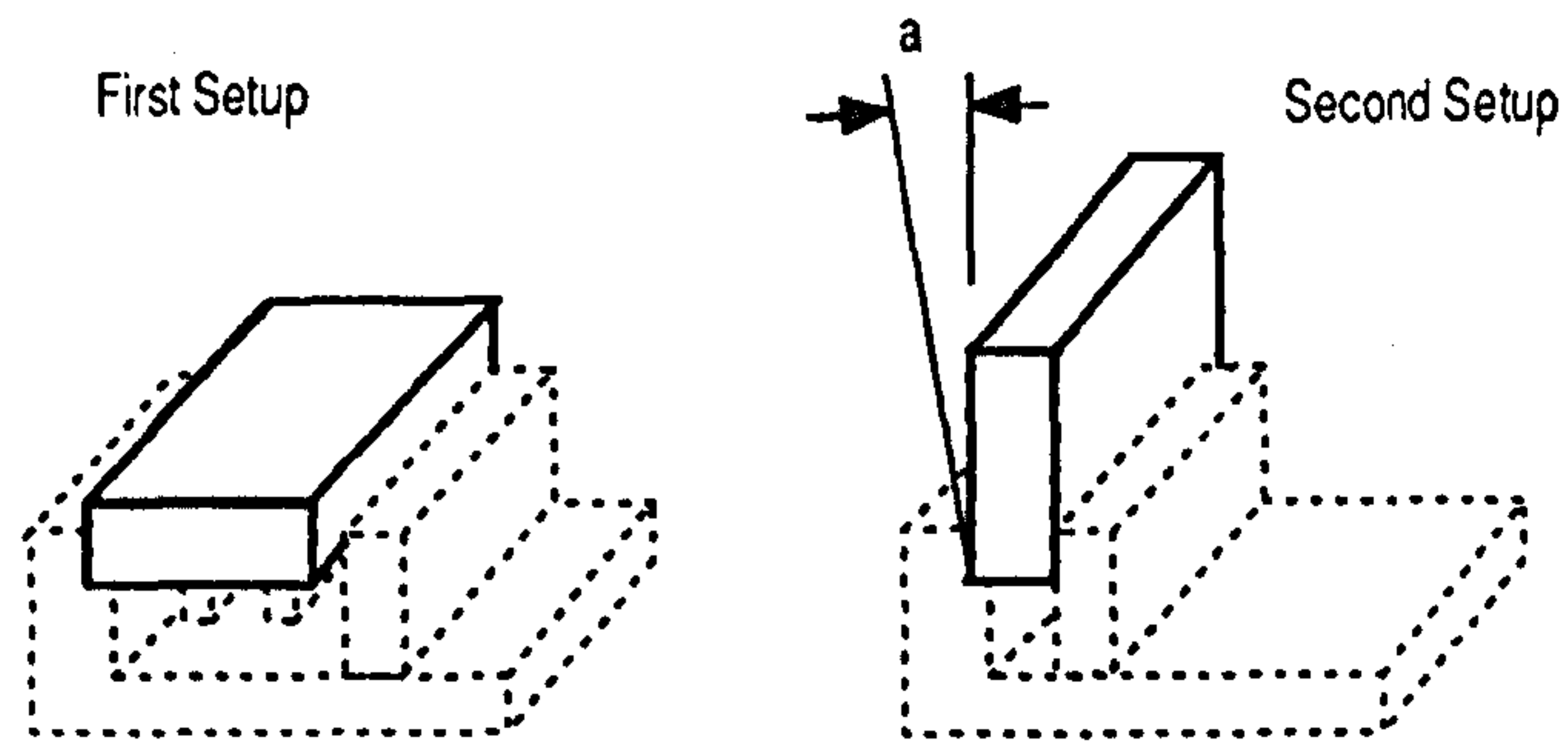
Figure 6.7(a) illustrates one of the accuracy problems explored by Boerma [17]. With this, he explains, that in the first setup hole A is realised at position  $Z(A)$ ,  $Y(A)$  of the machine tool coordinate system. In the second setup hole A is repositioned with an error 'd' in the Y direction, so hole A is at position  $Z(A)$ ,  $Y(A) + d$ . Therefore hole B, which should be machined at  $X(B)$ ,  $Y(B)$  is actually machined at  $X(B)$ ,  $Y(B) + d$ . It then depends on the value of 'd' whether the position of hole B will meet the requirements of the relation between A and B. This particular problem can be overcome by using a datum on the component and a touch trigger probe to reference the datum point. However, this may not always be possible and the orientation of the component with respect to the machine table may still introduce alignment errors. Dimensional relationships are therefore an important consideration in setup planning.

The shape of the workpiece is significant in providing a stable platform to fixture to the machine table, ensuring minimal misalignment of the workpiece as illustrated in figure 6.7(b). The accuracy of alignment will effect the geometric tolerances, as well as the dimensional tolerances, which can be achieved. For example, squareness could be a problem in figure 6.7 (b).

Technological constraints also apply, for example, the amount of material to be removed from a workpiece can influence the manufacturing method. Large material volumes can result in distortion of the component. It is therefore important to remove such material early in a setup sequence so that effects of distortion can be minimised. Such constraints on the methods to be used must be captured.



(a) Errors in Feature Position [ref. 17]



(b) Errors in Alignment

### Accuracy Problems in Setups

Figure 6.7

The ways in which fixtures can be used to hold workpieces leads to the idea of representing alternative methods as fixturing strategies. The Quick Turnaround Cell work [76] explored the use of a vice as a particular method of fixturing while Willis [149], in considering different setup strategies, identified the need to consider the global geometry of the component and geometric constraints on the machining of the part.

In order to explore the links between product data and setup planning the author has chosen to consider three fixturing strategies based on the use of a vice, the use of bolts and the use of clamps. These strategies have been used to represent alternative ways in which a machine, in combination with fixtures, can be used to hold a workpiece for machining. The author's work uses these strategies in the Machine Planning implementation, to demonstrate how product data, in conjunction with links to an SDSM Application, can provide necessary input to a Closely Coupled Machine Planner. A method of capturing geometric interactions in Fixturing Strategy selection is argued in chapter 7.4, while the experimental implementation used to describe and select fixturing strategies is described in chapter 9. It will be shown, through the experimental work described in chapter 11.3.4, that the approach taken provides a successful method of linking feature interaction checks into fixture strategy selection.

The use of a particular fixturing strategy has implications for the datum points to be used in each setup. Hence it seems appropriate to capture datums within the fixturing strategy. The way in which this has been achieved is explained in chapter 9.2.3 and 9.3.2.4. Selecting a datum has implications for the dimensional accuracy which can be achieved in the production of the component. The tolerance relationships on the designed component and the tolerance relationships which will result from the use of a particular strategy must be considered before a strategy can be confirmed as appropriate. How this could be accommodated in a Product Modelling



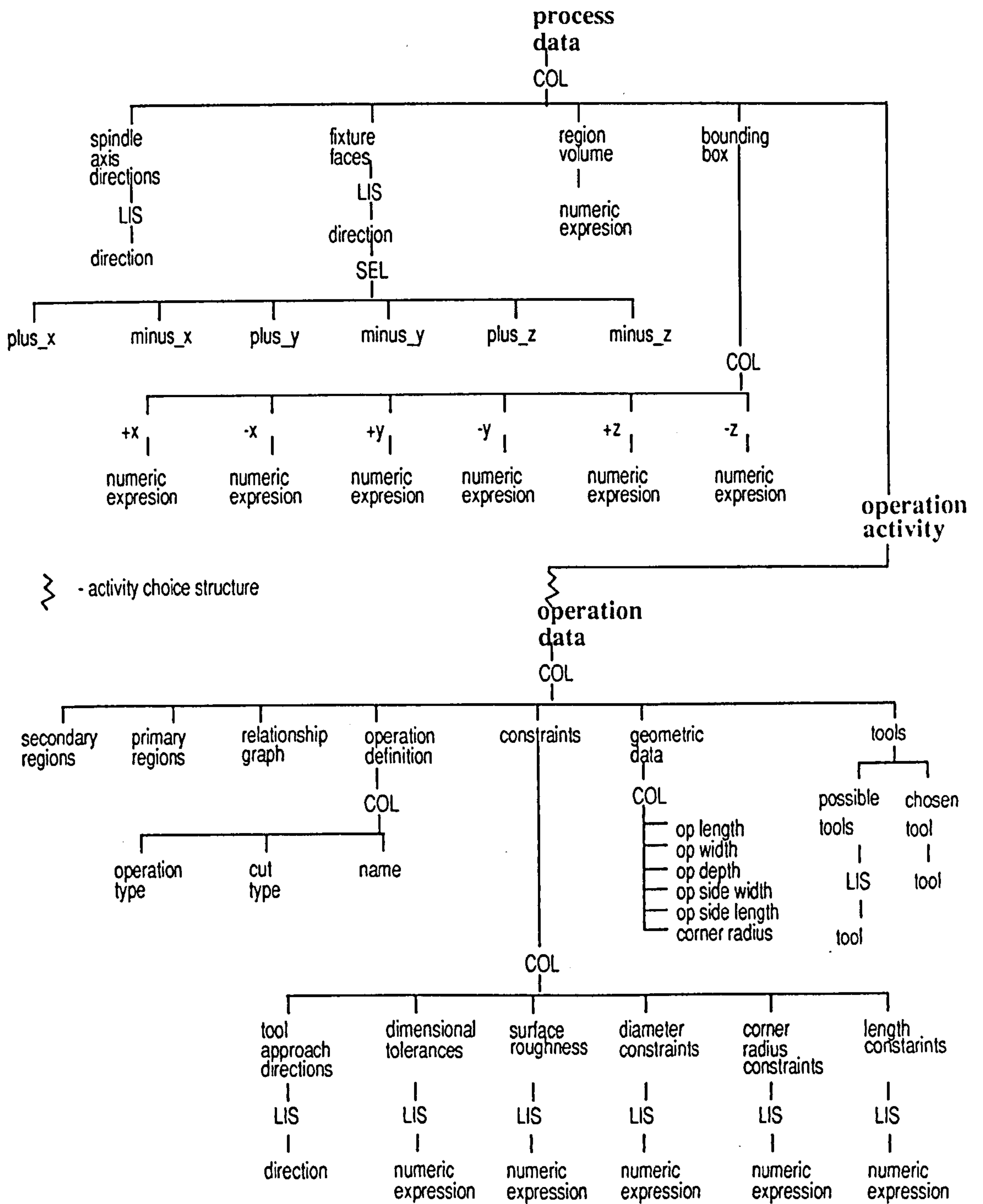
environment is discussed in chapter 8.

Once a fixturing strategy has been identified, and hence a setup sequence, each set-up must then be checked to ensure that all operations can be machined. It is possible that the fixture may cause cutter accessibility problems and if this is the case then changes in the machining plan must be made. The selection of *possible tools* as discussed in section 6.3.1 must therefore be performed during setup planning.

The structure and relationships between Machine Planning activities has been explored using the IDEF0 modelling technique. The resulting model is provided in appendix 3, and has been used as a basis for the software implementation described in chapter 9. The following sections argue structures for the lower levels of a Product Data Model, which have been used in the integration of Machine Planning, while chapters 9 and 10 describe the way in which an experimental Closely Coupled Machine Planner has been realised. These provide an insight into how future CAD/CAM systems can be radically improved by the use of such an integrated approach.

#### 6.4 PROCESS DATA RELATED TO DISCRETE MACHINING FEATURES.

The data associated with a machining feature provides constraints on how the feature should be produced. This data can be divided into general process data about the feature and the operation data which constrains how cutters can be used in the feature's production. Features are generally machined in more than one operation, Therefore a sequence of operations are related to a feature's process data as an *Operation Activity* of the sequential set type described in chapter 5.4.1. The feature level *process data* sub-set of the Product Data Model is illustrated in figure 6.8, while its composition is explained in the following sections.



**The Process Data Structure for a Machining Feature**

**Figure 6.8**

#### 6.4.1 General Process data.

The general process data to be associated with a machining feature, apart from the *Operation Activity*, has been identified as comprising the spindle axis directions, the fixturing faces, the region volume and the bounding box. This is the data which is generally applicable to the feature and not specifically related to any one operation. The identification of this data has followed from an assessment of the data requirements of Machine Planning and the results of experiments using the evolving Product Data Model as described in chapter 11.

The *spindle axis directions* identify the orientations from which it is possible to machine a feature with respect to the machine spindle axis. This can be represented as a simple list of directions, with one element for each possible direction. A machining feature contains, within it, faces which could be suitable for use in fixturing. These can also be referenced as a list of directions, and are stored as a list of directions, normal to appropriate faces, in the *fixturing faces* field.

The *region volume* field provides a value for the volume of the feature. This, when summed with other region volumes of features to be machined in a setup, provides a value for the volume to be removed in a setup. The *bounding box* of a feature provides data on the extremities of the feature's faces.

These elements of the process data structure provide the necessary data inputs, from a machining feature, required as an initial basis for setup planning. The use of this data structure, plus the *Operation Activity* structure described below, to build features which have been used in the experimental work of the thesis, is described in chapter 10.4.



### 6.4.2 The Derivation of Operation Data.

This section describes the view taken by the author of constraints on discrete shape machining which resulted in the operation data sub-set of the Product Data Model. This sub-set can be considered as being composed of three levels of refinement of data: the general geometric and dimensional data; the assessed geometric and dimensional data, identifying constraints on the operation's production method; and finally the tools which can be used to produce the operation. The following paragraphs briefly describe these three levels of refinement while the following subsections concentrate particularly on the author's contribution to the representation of the middle level of data which encompasses operation definition, constraints and geometric data.

The first level of data describes the geometric and dimensional data related to the operation. These are represented as primary regions, secondary regions and relationship graph. The primary regions represent the geometry which must be removed to machine the operation. The secondary regions represent the geometry, if any, which may also be machined as a consequence of machining the operation. This terminology has been coined by Bell [12]. The Relationship Graph of Wickens [148] represents dimensional relationships. This level of data provides the base level of information about an operation.

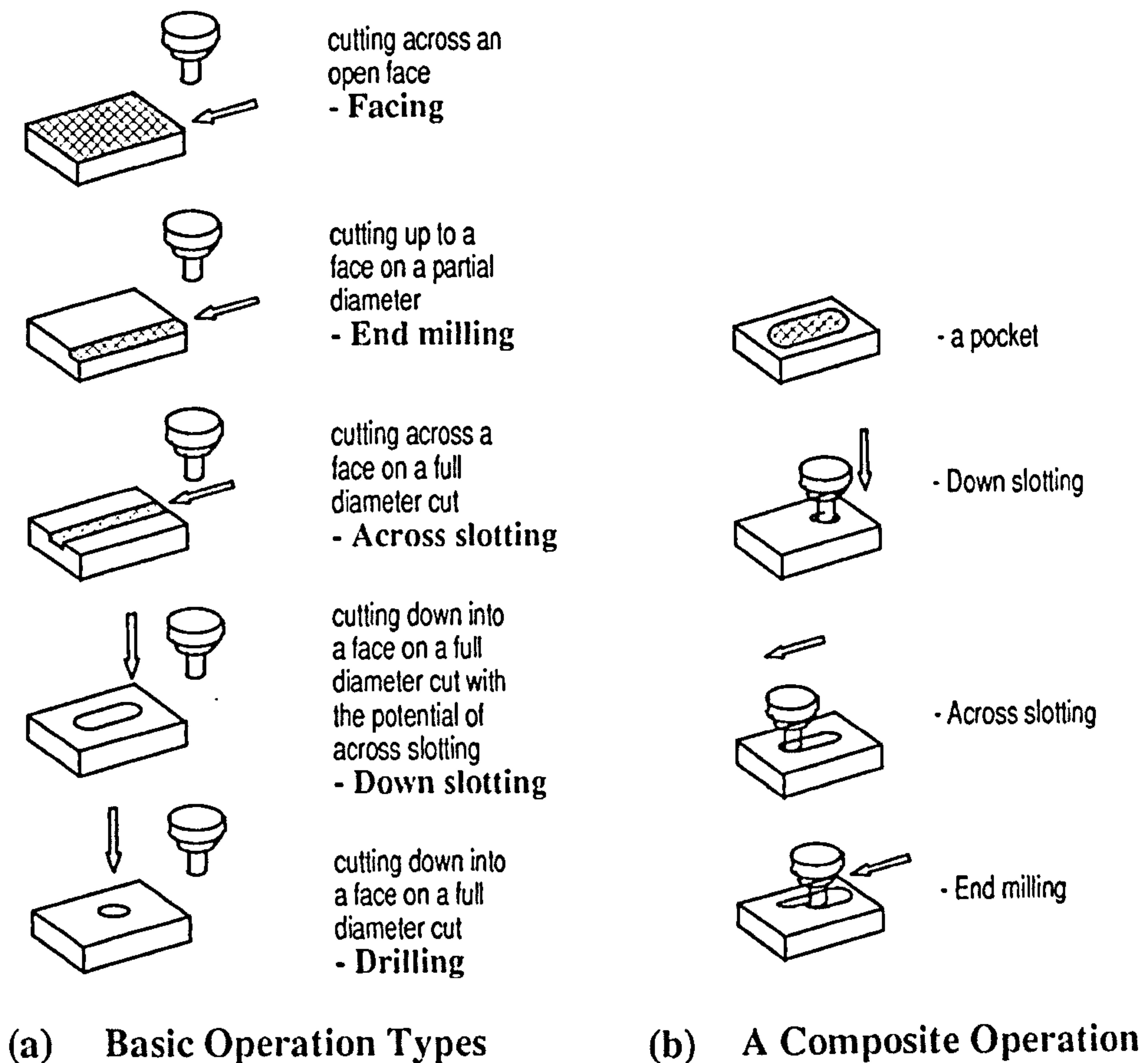
The second level of data describes the constraints on an operation's production method. This provides the significance for manufacture of the operation. This level of data has been divided into operation definition data, tooling constraints, which introduce restrictions on tool usage, and geometric data, which is refined from the first level of data to provide the significant geometric parameters for the operation's production.

The third level provides data on the possible tools which could be used to machine the operation and the tool selected for use. This data includes the cutting parameters to be used with each tool. This third level of data is generated by a Machine Planner by analysing the data at the second level. The implementation of functions to perform this level of planning are described in chapter 9.3.3.1 and 9.4.

### 6.4.3 Operation Definition Data from Machine and Cutter Combinations

This research proposed that the accessibility of the shape geometry should define the operation type, which through the use of technological rules can then provide a basis for cutter type selection. Consider figure 6.9(a). Here, five types of operation are defined; Facing, End Milling, Across Slotting, Down Slotting and Drilling. These follow from the accessibility of the geometry to be machined, as can be seen from the figure. Facing operation geometry can be accessed from any direction perpendicular to the spindle axis. End Milling operations can be accessed from two directions perpendicular to the spindle axis, and also have no constraint on the cutter diameter. An Across Slotting operation can be accessed from at least one direction perpendicular to the spindle axis and also constrains the cutter such that the full width of the tool must be used when machining. Down Slotting operations are accessed in the same direction as the spindle axis, but have the potential to be machined perpendicular to the spindle axis thereafter. Drilling operations can be machined only in same direction as the spindle axis.

When geometry, such as a pocket illustrated in figure 6.9(b), is considered, no one of the above operations describes the type of operation required. The machining of a pocket requires the combination of Down Slotting, Across Slotting and End Milling. The Composite operation type has been introduced to allow for this sort of occurrence. The Composite operation type consists of a list of the operation types required to machine the geometry in question.



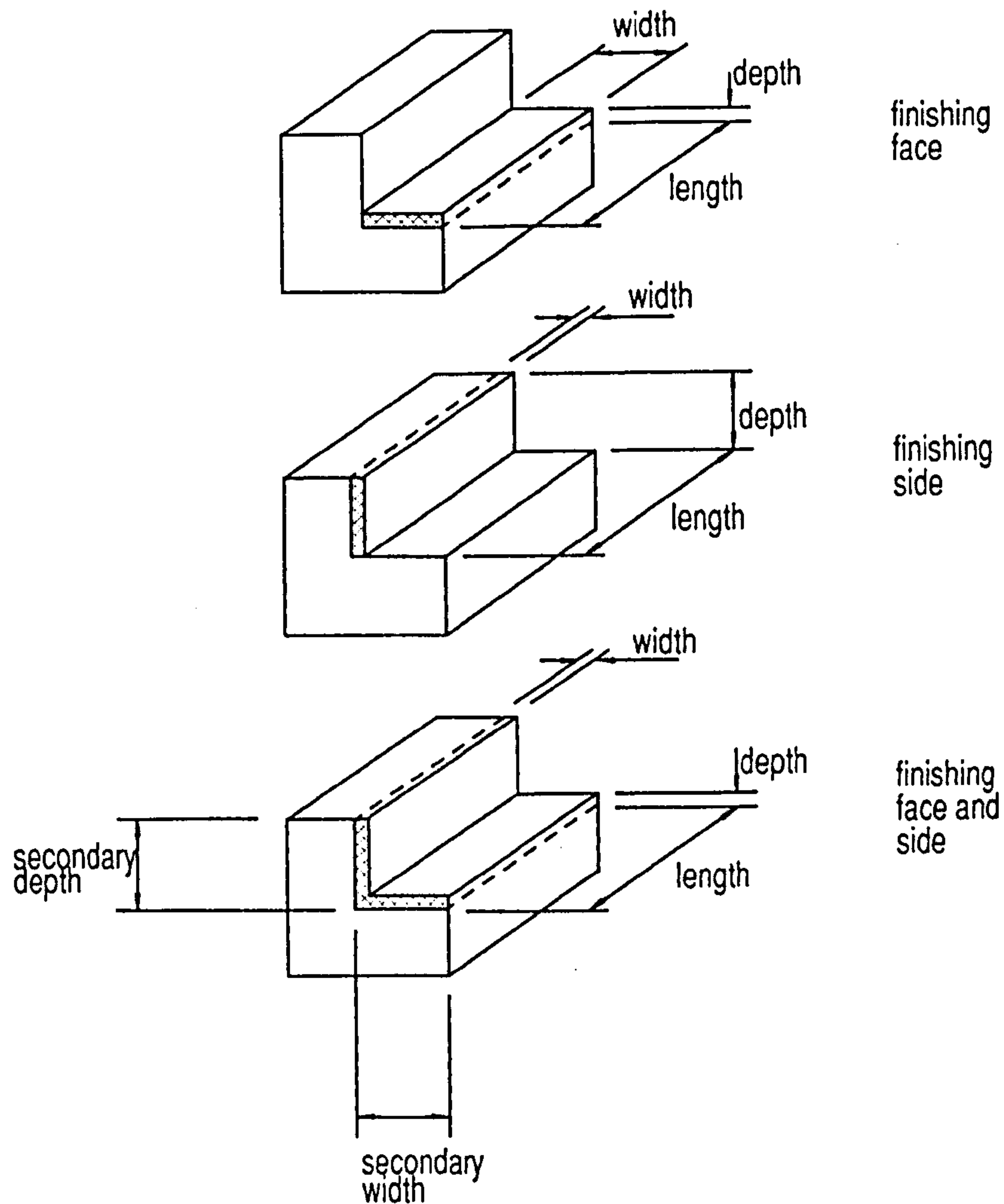
## Operation Types used in the Research

Figure 6.9

When rough machining a component the general aim is to remove the required material from the workpiece as quickly and effectively as the tooling capability will allow. However, when finish machining a surface it is normal to limit the cut being taken to minimise the cut load effect to ensure that the surface is produced within the required tolerances. This difference in requirement between operations can be identified by defining a cut type to flag the relevant information. Hence cut types for roughing and finishing are required. Cut type is also influenced by the geometry to be machined. Figure 6.10 illustrates the finishing operation terminology used by addressing a shoulder, which can be finished in three different ways, dependent on the required tolerances; either by finishing just the face, finishing just the side or



finishing both the face and the side. Hence these three alternatives are included in the cut type options.



### Finishing Operation Terminology

Figure 6.10

The final field which identifies the definition of an operation is the type of tool path which is to be used in its machining. A reference to the type of tool path to be used provides the basis to call the required tool path generation algorithms to generate actual tool paths. Two routes can be used to identify this information: algorithms to automatically generate cutter paths from the operation geometry [11]; and parametrically defined tool path plans which can be associated with specific operation shapes to generate the path [34]. Part of the work of this thesis has been

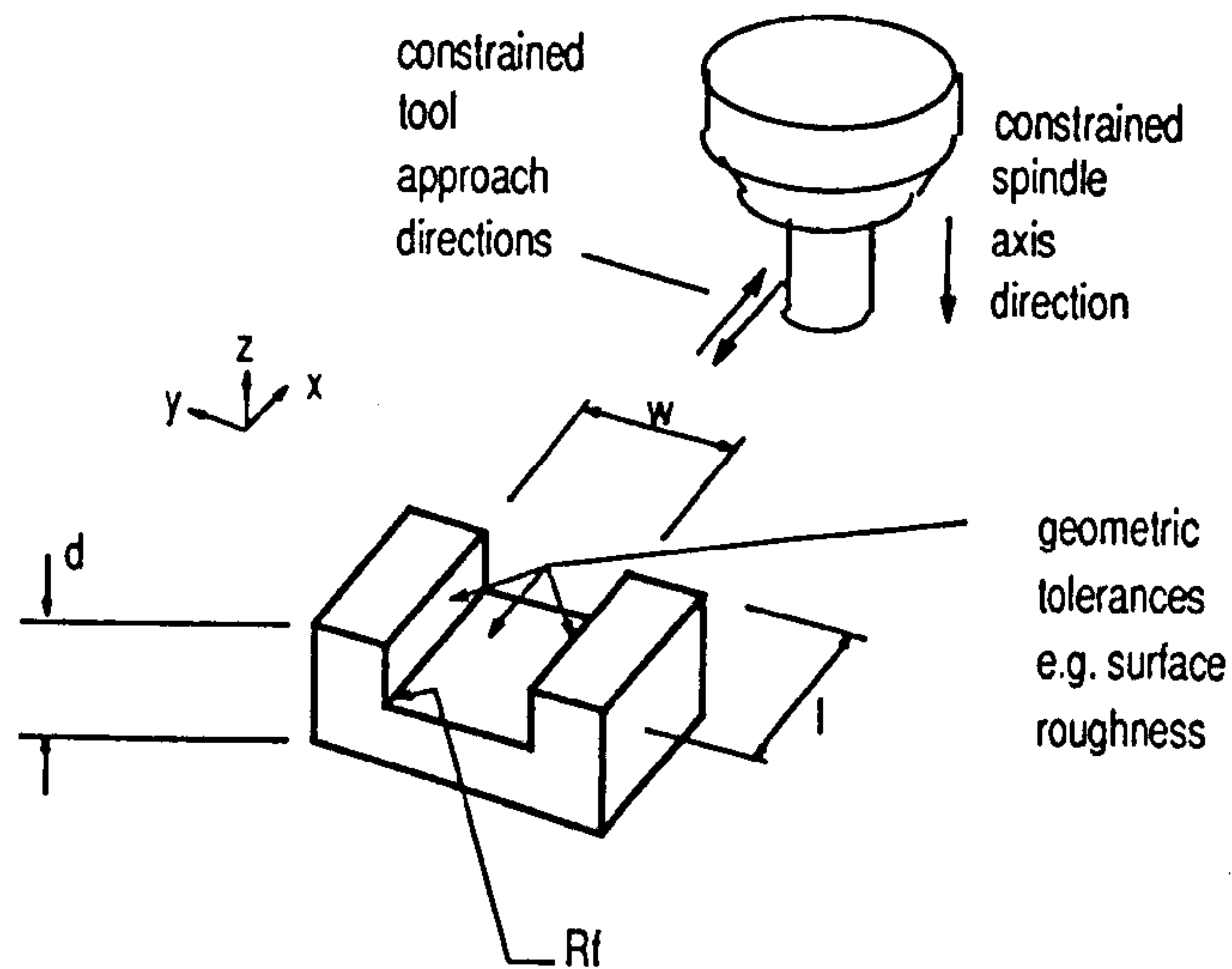
linked to both approaches, although the major implementation of experimental software has been pursued following the tool path plan route, as sets of appropriate tool paths plans were configured to suit the Machine Planning requirements by M.J. Corrigall [34].

#### 6.4.4 Constraints on Operations.

Constraints on the machining of discrete shapes can come from the shape itself and also from technological rules on the appropriate use of tooling. The following paragraphs describe the various constraints represented in the operation data sub-set of the PDM. Figure 6.11 shows an example operation which illustrates the constraints.

When machining an operation the cutter must access the shape from some direction. The possible directions which can be used are termed the tool approach directions. For example the operation in Figure 6.11 has two tool approach directions perpendicular to the spindle axis while a blind hole has one tool approach direction parallel with the spindle axis.

Dimensional tolerances influence the cutting parameters to be used in machining an operation. Those tolerances which may be significant to the operation's production can be stored as a list of tolerance values in the dimensional tolerance constraint field. Tolerance values are defined as a combination of a nominal value and a tolerance. These tolerance values can then be assessed when calculating cutting parameters to ensure the operation is produced within specification. Storing tolerance values in this way means that they can be generated, either from feature parameters, or from a Relationship Graph evaluation.



Parameter list :-

$l$  ::= length (tolerance value)

$w$  ::= width (tolerance value)

$d$  ::= depth (tolerance value)

$R_f$  ::= Fillet radius (tolerance value)

$R_a$  ::= Surface Roughness (real number)

tool constraints:-

tool approach directions:  $+x, -x$

dimensional tolerances:  $w, d$

surface roughness:  $R_a$

diameter:  $w$

corner radius:  $R_f$

length:  $d$

### Constraints on Tooling from the Operation Data Definition

Figure 6.11

The surface roughness is another tolerance which influences the calculation of cutting parameters. Again the appropriate values can be stored as a constraint on the operation and used in procedures to calculate machining conditions.

Dependent on the shape being machining there may be no parameters, one, or many parameters, which constrain the diameter of the tools which can be used. Any parameter which may provide a constraint can be stored in the diameter constraint list and assessed by software procedures to determine the critical value. Similarly parameters which constrain the tool corner radius or the tool length can be stored in the appropriate constraint field.



#### 6.4.5 Geometric Parameters influencing Operation Machining.

When planning the machining of operations the calculation of cutting parameters and, in the approach taken, the generation of tool paths can be achieved by reference to the key geometric parameters of the shape. These key parameters can be stored in the geometric data section of the operation data sub-set. This avoids the need for unnecessary analysis of the operation geometry. The parameters stored with the operation data sub-set are as illustrated in figure 6.8 and are the operation length, width, depth, side width, side depth, and corner radius. These can all be evaluated directly from feature parameters.

#### 6.4.6 Cutting Tool Data Related to Machine Operations.

The data defined in the earlier sections provides the basis on which cutters have been chosen to machine an operation, as argued in chapter 6.3.1, using the implemented Machine Planner function described in chapter 9.3.3.1. Hence for any individual operation a list of possible tools can be chosen. A set of operations can be sequenced and a specific tool can be chosen to machine the operation, as argued in chapter 6.3.2 and implemented in chapter 9.4. The data structure for cutting tools provide the ability to capture these data interactions.

The feature level *process data* sub-set of the PDM will be shown, in the experiments described in chapter 11, to be a necessary source of data for the experimental Closely Coupled Machine Planner. This, in combination with the component level *Planned Processes* described in the following section, and the means to identify inter-feature data, as discussed in chapters 7 and 8, provide key product data sub-sets, necessary for a Closely Coupled Machine Planner.

## 6.5 PROCESS DATA RELATED TO COMPONENTS.

The planning of a component's manufacturing method involves a range of activities. Following the general integration argument proposed in chapter 5, which advocates the use of a product data model as an integration mechanism, there is a need for a sub-set of the Product Data Model to represent the relevant data structures to be used, to store data related to process planning functions. This sub-set of the Product Data Model has been represented by the component level *Planned Processes*, as illustrated in figure 5.3 and expanded in figures 6.12 (a) and (b). This data sub-set has been divided into four main levels to reflect the four main decision levels related to planning the manufacture of a component. These being the process, machine, setup and operation levels illustrated in figure 6.12. The data structure at each level is discussed in the following sections, highlighting their relationships to the integration of the various planning aspects of this research.

### 6.5.1 Planned Processes - Process Level Data.

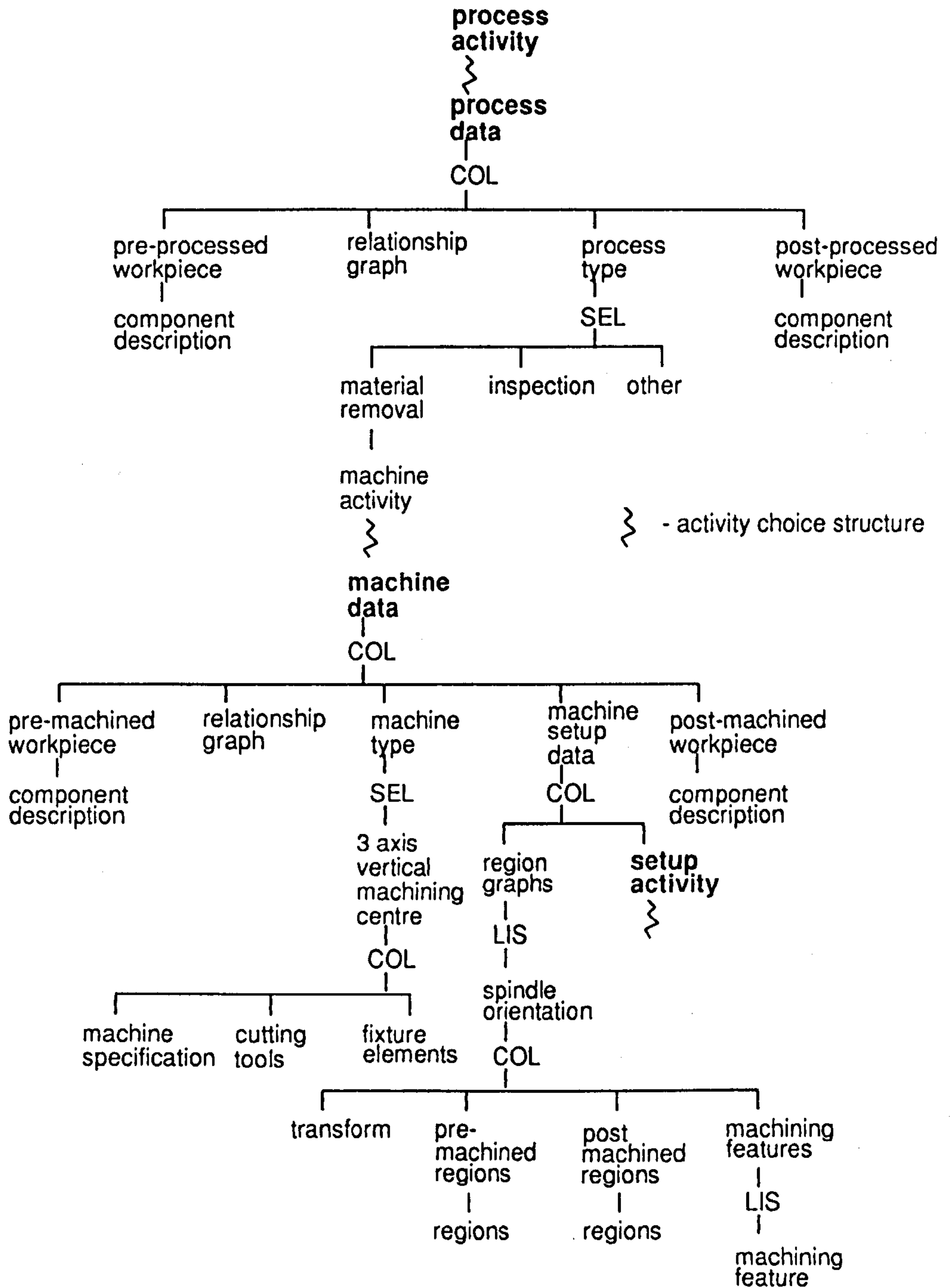
The consideration of alternative processes has not formed a part of this research. However, in defining an appropriate data structure it is important to acknowledge that the production of a component can take it through a number of processes. Hence the process level data has been included in the model. It is assumed that for any process there will be a start and an end condition for the workpiece. This is reflected in the *pre-processed* and *post-processed workpiece* fields provided in the structure. A *relationship graph* field [148] has been included at this, and every, level of the structure, to reflect the dimensions and tolerances which are applicable at that level. The *process type* field provides the key to the machine type being considered. In this case material removal and inspection processes are included to reflect the emphasis of the ISS project. A field for other processes is included to highlight that other processes should be included and explored.

### 6.5.2 Planned Processes - Machine Level Data.

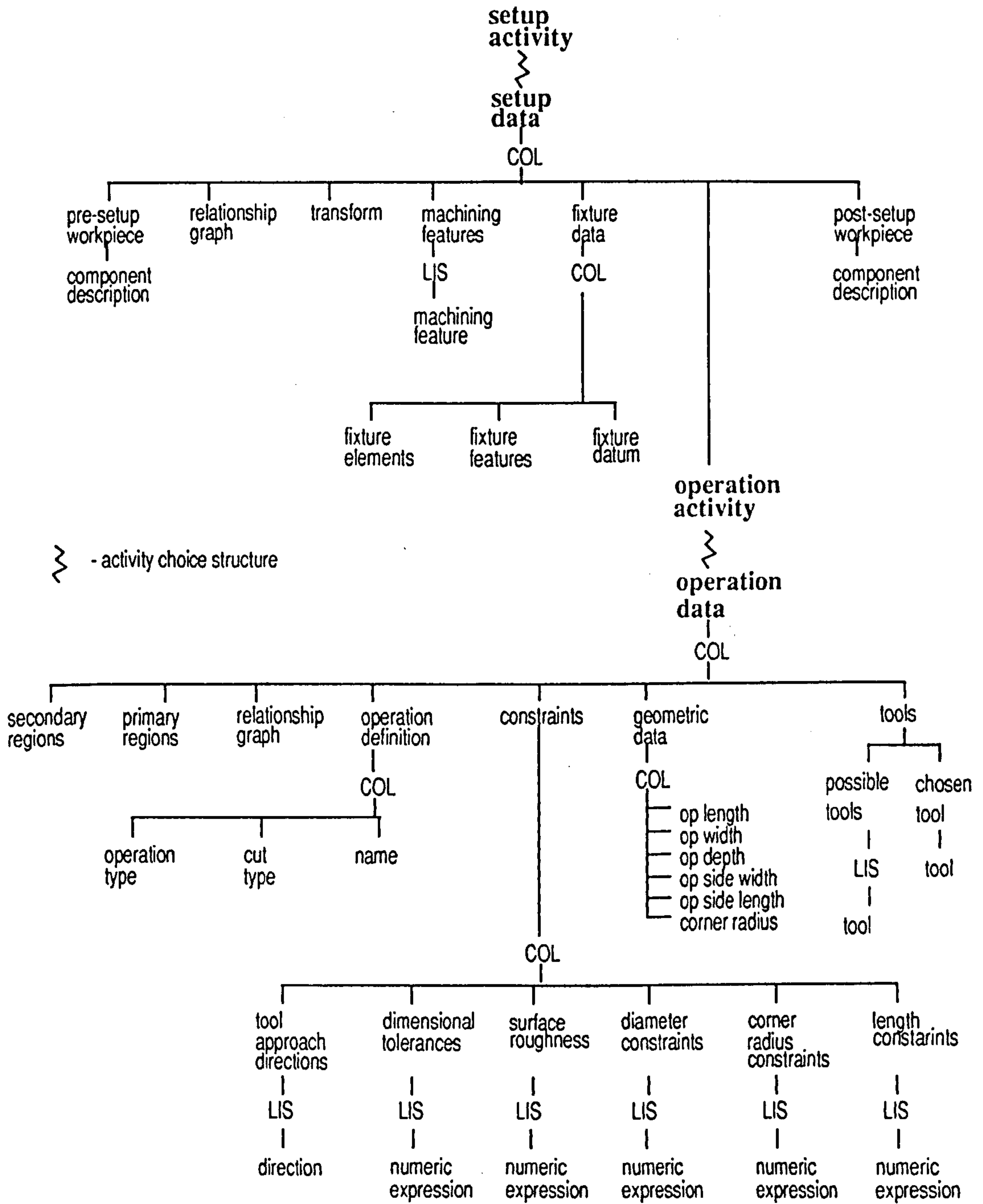
The *pre-machined* and *post-machined workpiece* fields provide for the geometry of the workpiece, as it comes to the machine tool , and as it leaves the machine tool. As this research is constrained to addressing components to be machined on a single machine, the *pre-processed* and *pre-machined workpiece* are the same and represent the stock to be machined. Similarly the *post-processed* and *post-machined workpiece* are the same and represent the component to be produced. The relationship graph, as before, represents the dimensions and tolerances of the workpiece.

Planning decisions on the product are dependent on the machine and the tooling being considered. Therefore a link to the *machine specification, cutting tools and fixture elements* which are used in Machine Planning are included at this level. The *machine setup data* provides the data on the regions to be machined on the component, as *region graphs*, and a *setup activity*. The former provides data for each spindle axis direction of the transform of the setup required to align it with the spindle axis, and the list of machining features which can be produced from that particular spindle axis direction. Additional inputs to the this aspect of the structure were made by P.Bell [12]. The *setup activity* stores data describing the setups to be used in producing the component.





**Process and Machine Data Structures  
for Planned Processes:  
Figure 6.12 (a)**



**Setup and Operation Data Structures for Planned Processes:**

**Figure 6.12 (b)**

The machine level data in the *planned processes* provides the basis for the link between pre-defined features and recognised features as illustrated in figure 6.13. The figure draws on the data structures provided in figures 5.3 and 6.12(a) while the bold lines indicate the flow of data from a Product Model to an Application and the flow of data from an Application to a Product Model.

Feature recognition would take as input the *pre-machined* and *post-machined workpiece* and the *relationship graph* to generate the data relationships for each spindle axis orientation resulting in the *machining features* list. Alternatively, where pre-defined features have been used, the machining features can be extracted directly from the component description and put in to the *machining features* list for each appropriate spindle orientation. In the case of combined pre-defined and recognised features, the pre-defined features can be put in to the appropriate list and the remaining geometry must be analysed. The Machine Planner Sequence Setups activity can then draw on the machine level data as shown in figure 6.13 to generate the necessary setup data in the *setup activity* data structure. This view of the data structure, as an integration mechanism for feature pre-definition and feature recognition has been developed in conjunction with P.Bell [95]. The exploration of the activity interactions to suit this integration concept are shown in the IDEF0 model of Machine Planning in appendix 3.

### 6.5.3 Planned Processes - Setup and Operation Level Data.

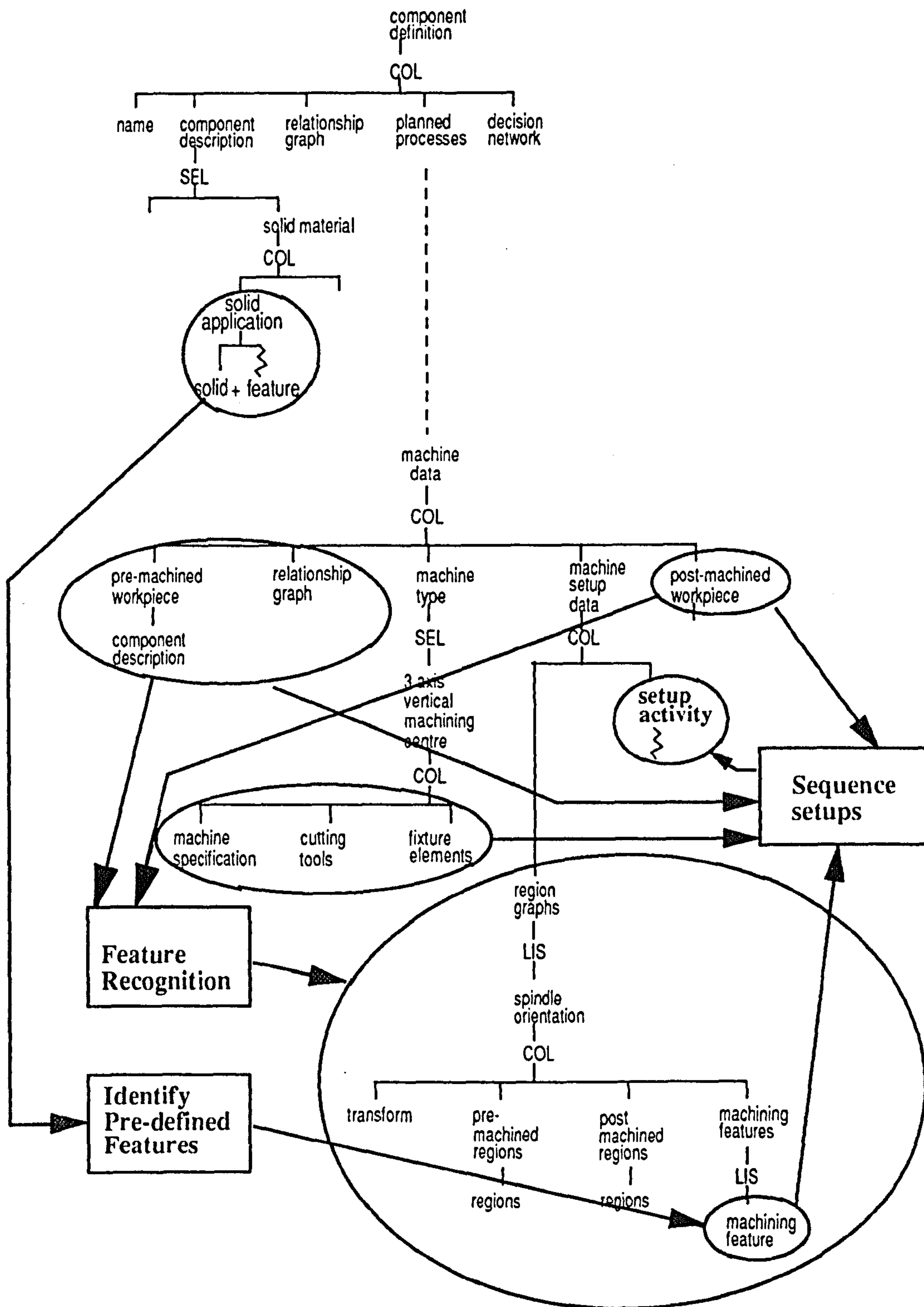
The setup and operation level data is illustrated in figure 6.12(b). The *pre-setup* and *post-setup workpiece* fields are provided to store the geometry of the workpiece as it comes to a setup and as it leaves the setup. These fields are particularly useful to the geometric analysis approach to cutter path generation of P. Bell [12], and provide the starting data for his work on the breakdown of the geometry into



operation data.

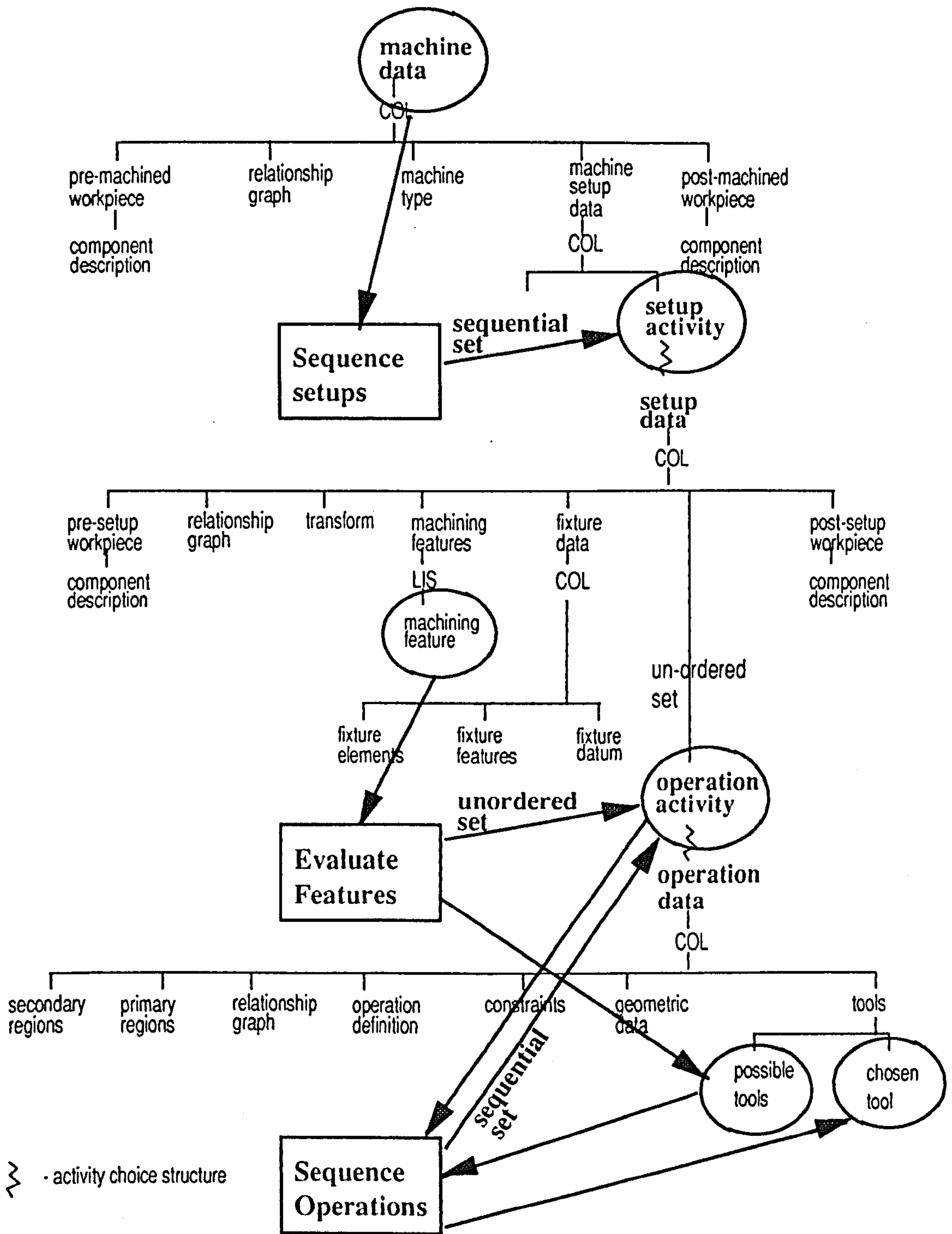
The integrating capability of the structure, for Machine Planning functions, is illustrated in figure 6.14. The Sequence Setups function takes data from the *machine data* level and provides data into the *setup activity*. The structure of *setup data* provides the fields to store the *transform* of the setup to align it with the spindle axis, the *machining features* to be produced in the setup, and the *fixture data* required for the setup. The *fixture data* contains the *fixture elements* to be used in the setup, the *fixture features* required to fixture the workpiece, and the *fixture datum* to be used. The Evaluate Features function can then evaluate the machining features to be produced in the setup. If these have been pre-defined, the operation Activity Data will already be available from each feature. If feature recognition is employed then operation data will need to be found. The *operation data* is the same structure as derived in the feature level structure.

The component level *planned processes* sub-set of the PDM provides another key link in the integration of manufacturing Applications, both in the ISS programme and as a source for future exploration of more comprehensive integrated design and manufacture systems. How the component level *planned processes* sub-set of the PDM has been linked to the experimental Machine Planning Application, is explained in chapter 10.5, while the description of its use in integration experiments is provided in chapter 11.4.



The Integration of Feature Identification Methods and Setup Sequencing using Planned Processes

Figure 6.13



The Integration of Machine Planning Activities using Planned Processes

Figure 6.14



**CHAPTER 7****THE INTERACTION OF GEOMETRIC AND MACHINE PLANNING  
APPLICATIONS IN A PRODUCT MODEL ENVIRONMENT****7.1 INTRODUCTION**

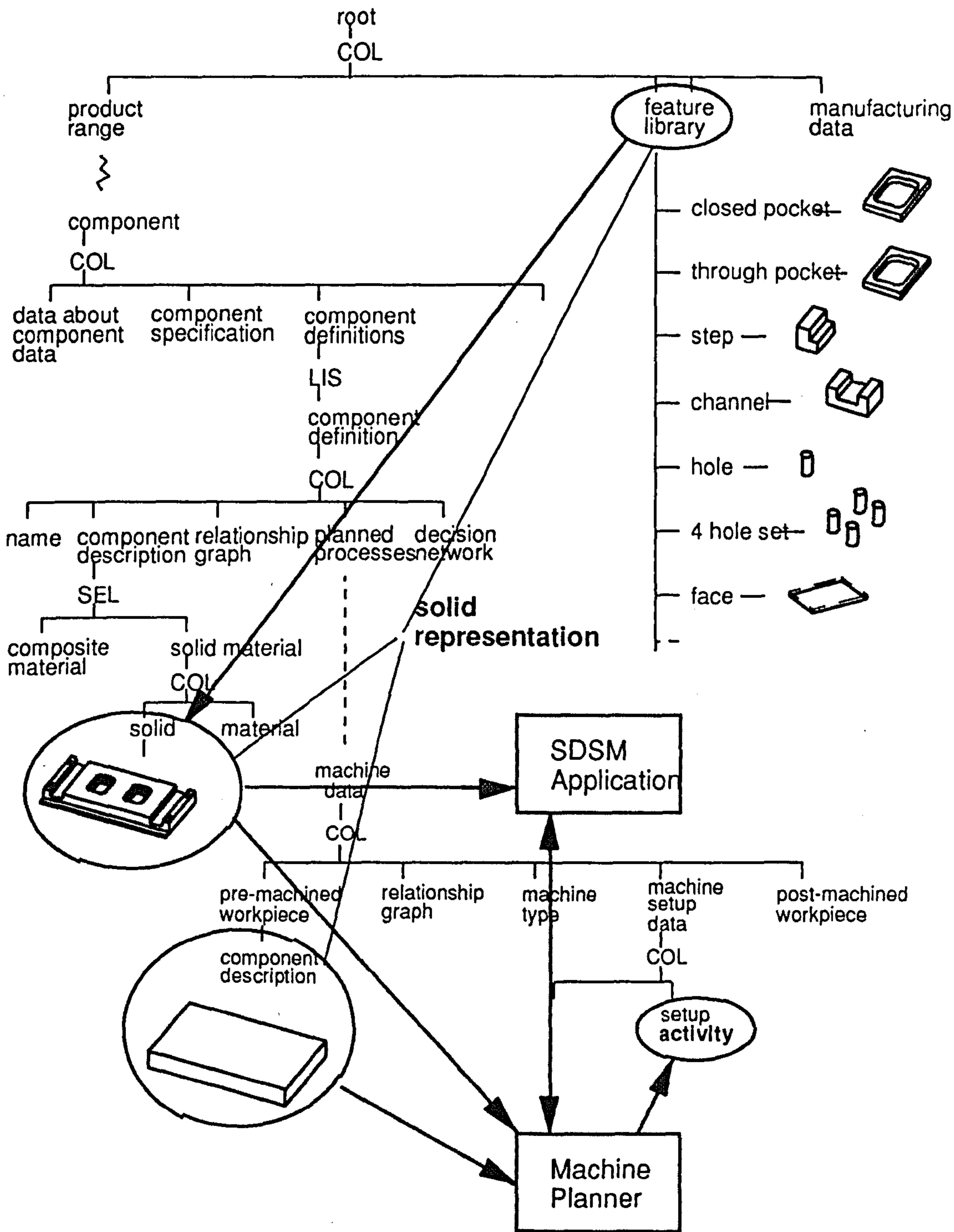
This chapter investigates the benefits which the Product Model environment can offer to the interactions between geometry and Machine Planning Applications, and describes the methods used to derive, through a PDM, feature positional and geometric interaction data. Links have been achieved to generate this data using the experimental Machine Planner described in chapter 9, and the success of the approach taken is explored in the experiments discussed in chapter 11. Elements of the experimental software are described in appendix 5.

Chapter 6 has established the Product Data requirements for Machine Planning and identified the need for feature position and geometric data. The methods described in this chapter assume that a component model will be populated, as described in chapter 10, to provide a Product Model with which a Closely Coupled Machine Planner can interact.

## 7.2 GEOMETRIC AND MACHINE PLANNER INTERACTIONS

A Machine Planner, whether stand alone or integrated, requires geometric data to function effectively, as discussed in chapter 6.2. The Close Coupling of a Machine Planner opens up the possibility of linking to the geometric Applications available in the Product Model environment, as illustrated in figure 7.1. This figure shows that a solid representation of a component, and its related stock, can be built using features from a feature library. The solid representation produced, based on CSG, provides a geometric input to a Machine Planner of the component and the stock. A second geometric Application, the SDSM, has been used to evaluate the solid representation of the component, to generate a cell decomposition model. This additional geometric representation provides the basis for the author's exploration of feature interactions, described in section 7.4.

In addition to using geometric data, planning functions should be able to interact with geometric Applications to update and change the geometric description contained in a Product Model, enabling other Applications to then interact with the newly generated product data. An example of a Machine Planner influencing the geometry in a Product Model can be seen by considering the setup data sub-set of the PDM, as illustrated in figure 6.12(b) In this figure a pre-setup and post-setup workpiece is represented, describing the geometry before, and after, the setup. This data should be generated by a setup planning function in a Machine Planner. The resulting geometric description could then be used in the analysis of setup specific data, by Applications such as that of P. Bell [12].



Geometry and MP Interactions

Figure 7.1



Links to two geometric representations, through the PDM, have been implemented by the author. These links enable the generation of feature position and feature interaction data, required by the Machine Planner. The detailed aspects of a method to identify the relative feature positions needed in Machine Planning, and a method to identify feature interaction data, are described in the following sections. The implementation of the experimental Machine Planner which used these methods is described in chapter 9, while the experiments performed, demonstrating the success of the approach taken, are described in chapter 11.

### **7.3 THE METHODS USED TO IDENTIFY FEATURE POSITIONAL DATA**

The methods argued in this section illustrate how feature positional data can readily be derived, when working in a Product Model environment. How product data can be manipulated to provide appropriate inputs, for setup planning and NC code generation, is described. This emphasises the value of the PDM in providing a ready source of data which can be used, and manipulated, by a Closely Coupled Machine Planner.

#### **7.3.1 Feature Positions on a Component.**

The relative position of a feature on a component needs to be measured with respect to the component, rather than from some global coordinate system, if its position on the component is to have some meaning for planning. This requirement has been identified by Kanumury [76]. Part programs for machining rely on knowledge of the positions of machining operations relative to a fixture datum and hence this data must be extracted from the PDM. The component coordinate system has been taken to be the centre of the component block to which features are

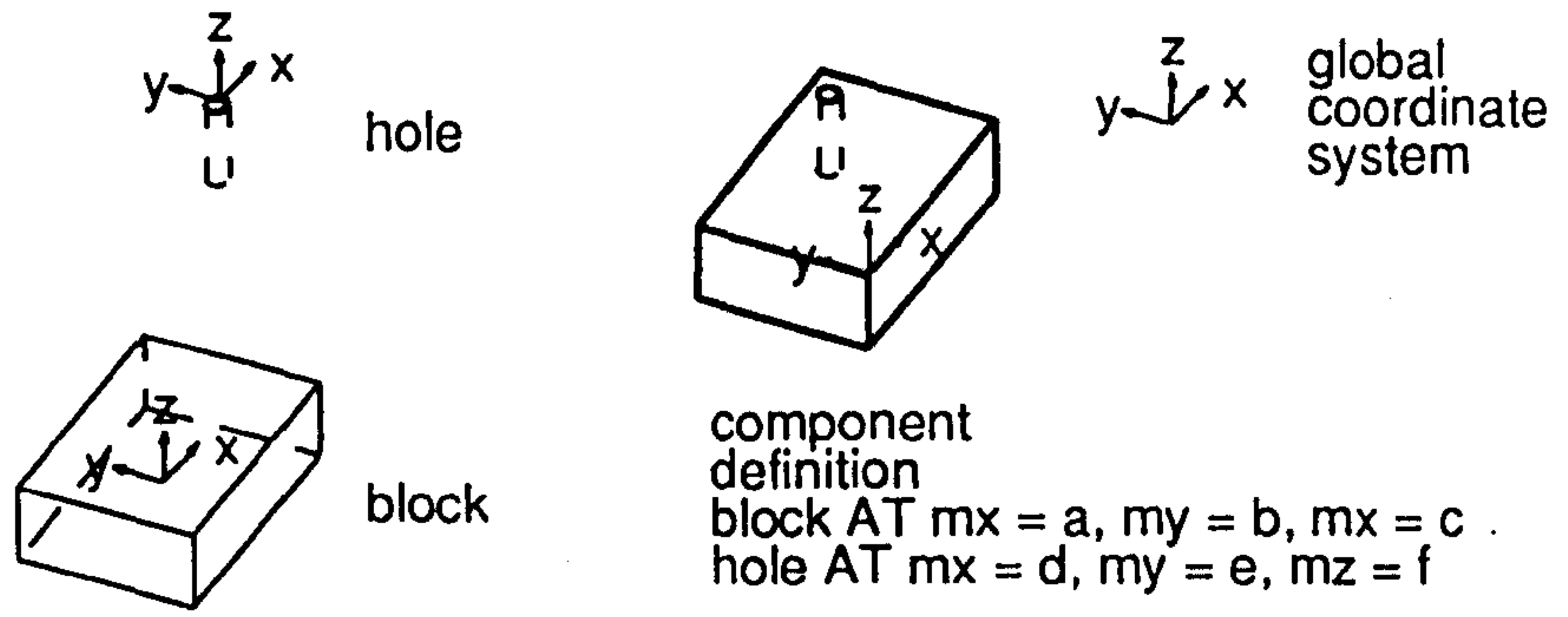
added. This means that feature positions, relative to the general geometry of the component, can be found.

The reference position of a feature definition is at (0,0,0) in the global coordinate system as shown in figure 7.2(a). Likewise the centre of a block is defined at (0,0,0). Hence the position of a feature on a component can be found readily from the transformations of the component block and the feature, when they are used in a component definition. This can be captured simply, as shown below, and has been shown to be successfully applied in the experiments described in chapter 11, using the Machine Planner implementation described in chapter 9.3.1

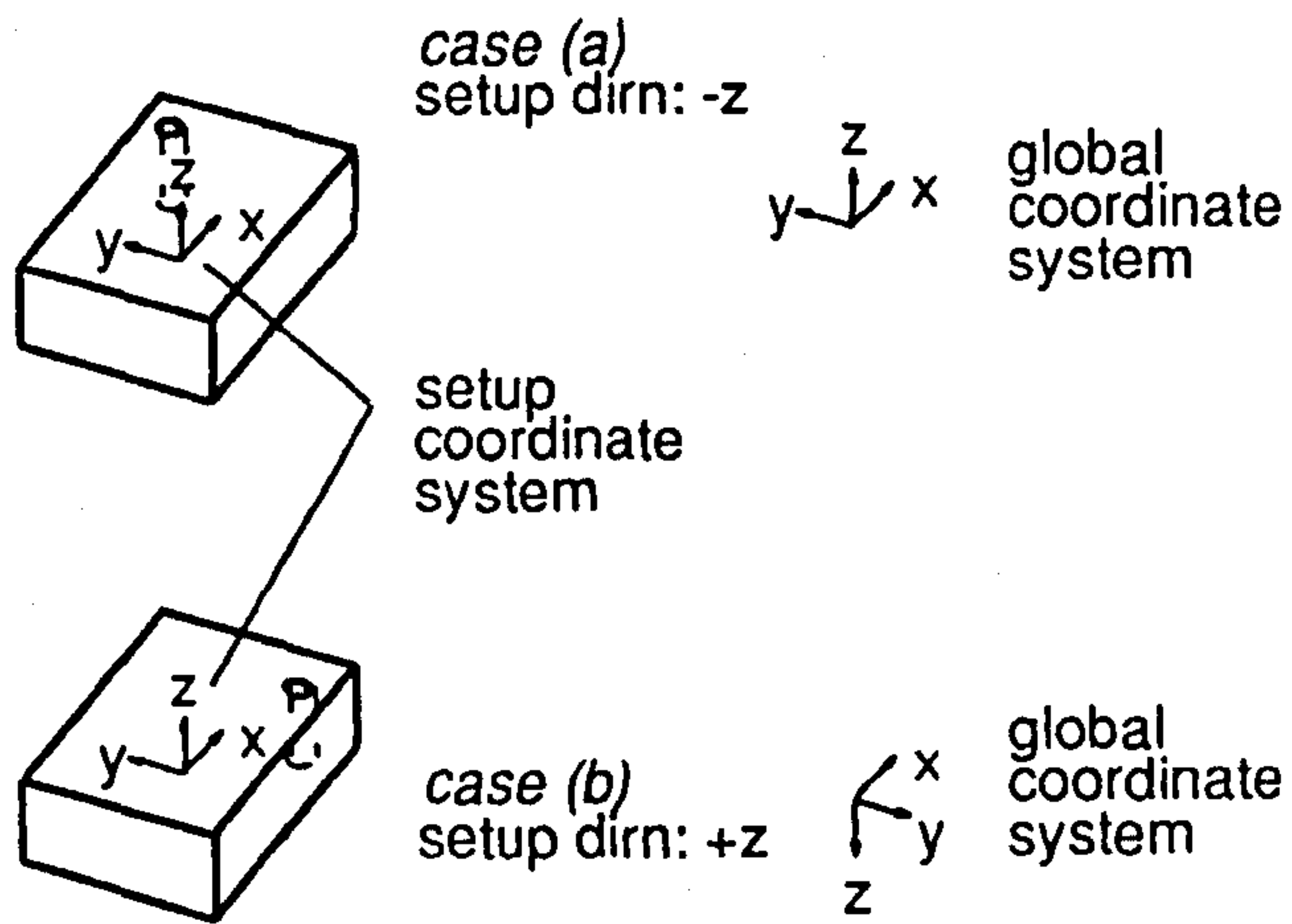
$$\begin{aligned} \text{feature position with respect to component} &= \\ \text{feature global position} &- \text{component global position} \end{aligned}$$

The relative position of a feature within a setup is different from that relative to the component, and is dependent on the setup direction. As can be seen in figure 7.2(b), the position of the hole in case (a) is (x,y,z). When the component is rotated about the x axis to consider a setup in the +z direction, as shown in case (b), the hole position with respect to this setup is (x,-y,z). Hence, for each setup the positions of features has been reassessed, with reference to the setup coordinate axes. The setup coordinate axes are assumed to be the same position as the component coordinate axes, but with the axes adjusted, such that the setup z axis is opposed to the spindle axis direction. This will be shown, in the experimental work, described in chapter 11.3.3, to be a successful approach.

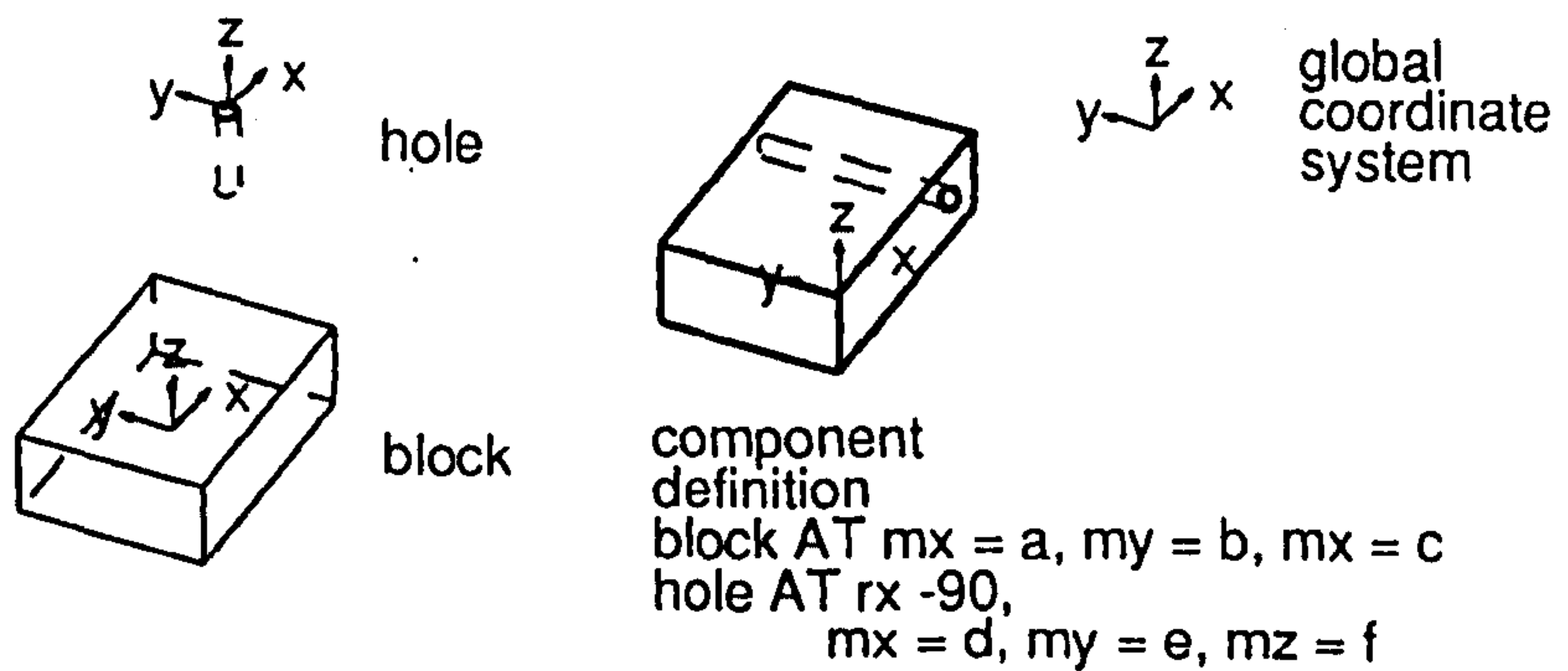
Datum features have been discussed in chapter 6.2.1 and can be represented within fixturing strategies, as will be explained in chapter 9.2.3. In order to generate part programs, there is a need to identify datum positions with respect to the machine table, and also to identify machining feature positions with respect to a datum.



(a) Feature positions on a Component



(b) Feature positions in a Setup



(c) Feature Orientations on a Component

Feature Positional Data

Figure 7.2

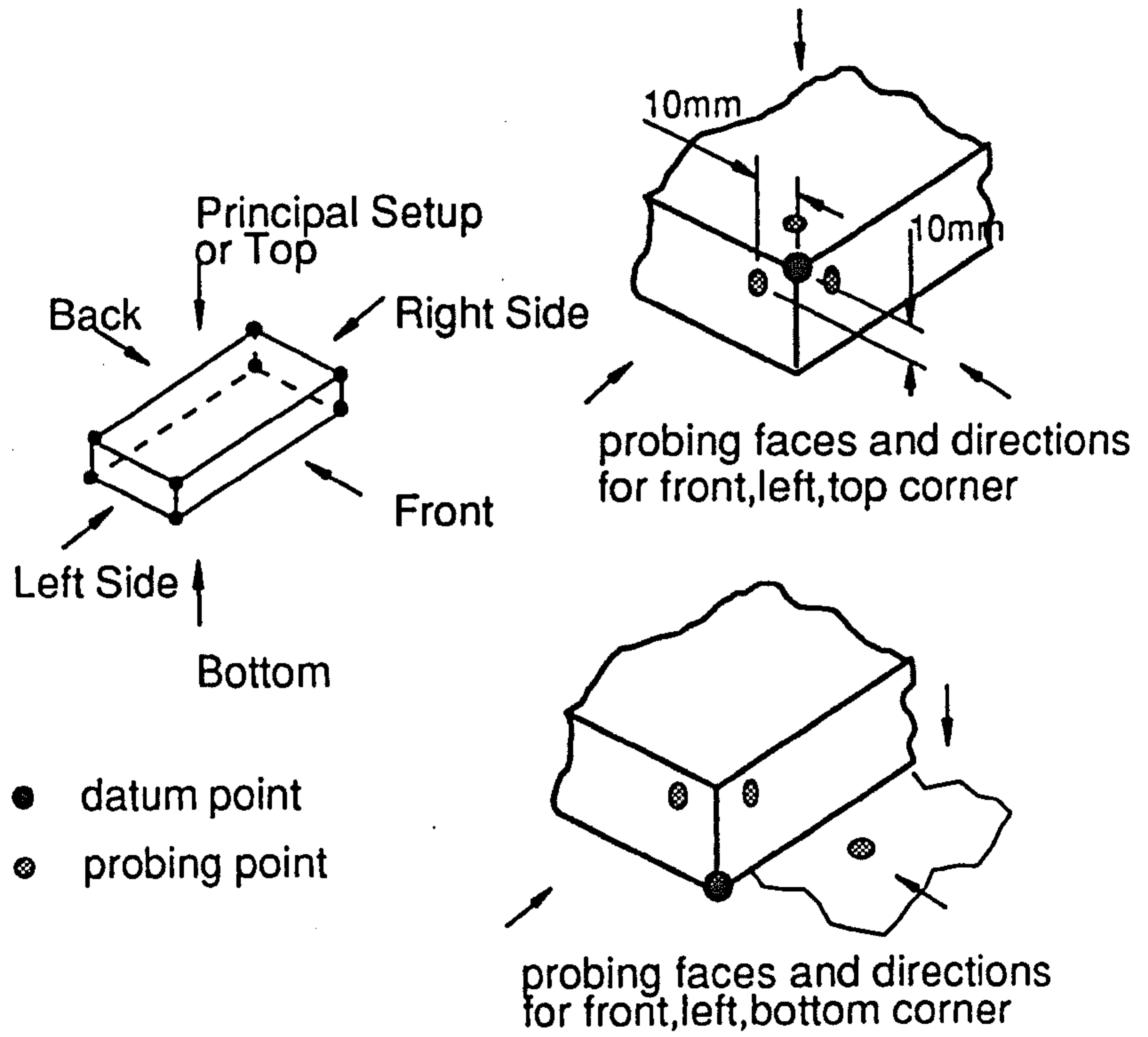


The fixture datum coordinates of an external corner can be found from the intersection of three faces. The coordinate value for each axis, relative to the machine table can be found by using a touch trigger probe and probing a face lying in the plane of the other axes. There are eight types of external corner, as shown in figure 7.3(a); four on the top surface and four on the bottom. The direction in which to probe follows from the corner type. In the case of corners on the top surface, the position on the surface to probe has been identified as a point relative to the datum coordinates. This has been taken as being 10mm onto the face to be probed, as illustrated in figure 7.3(a). However, when the bottom surface is used, an appropriate external datum surface, relating to the fixture, must be used. The probing position of this surface must be requested from the user, as such data is not captured in the product representation. The above approach has been implemented as a part of the *Extract Machine Control Code* function described in chapter 9.5, and has been used successfully in producing components on the Wadkin V4-6 machining centre, as described in chapter 11.3.3.

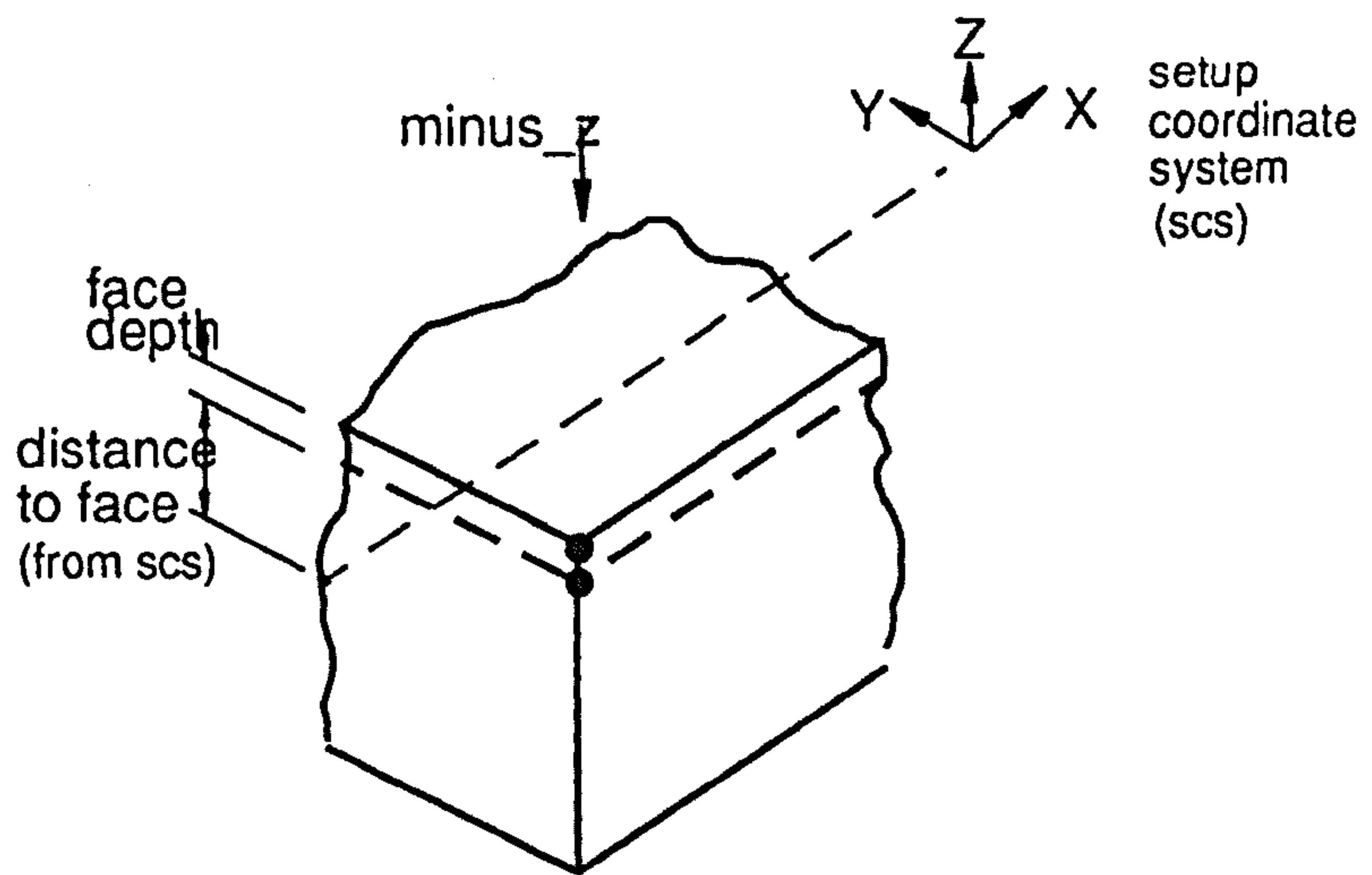
Machining feature positions with respect to a fixture datum have been found from the relation of the feature position in the setup to the datum position in the setup as:

$$\begin{aligned} \text{feature position with respect to datum} &= \\ \text{feature position in setup} &- \text{datum position in setup} \end{aligned}$$

The features used for fixture datums, in the software implementation, are holes and faces. When holes are used as datums they must already have been machined. However, when faces are used they may, or may not, have been machined. Hence, the coordinates of the fixture datum, relative to the setup coordinates, are dependent on whether the faces used in datum setting have been machined.



(a) External Corner Datums



(b) Machining effect on the datum position

Positional Data in Fixture Datum Setting  
Figure 7.3

The dependency can be seen from the illustration in figure 7.3(b) which shows two different relative positions, in the z direction, for the same datum corner. The same dependency also applies in the x and y directions. This difference in position can be accommodated in the software by checking whether the face features around the corner will have been machined before datum setting. For example, considering the case in figure 7.3(b), with a negative z probing direction then the z coordinate of the datum can be found using the conditional statement below:

```

if reg.machined (a machined feature)
then
  datum_coords(z) := setup_coords(z) + (part.geom.depth.num_value / 2.0);
else
  datum_coords(z) := setup_coords(z) + (part.geom.depth.num_value / 2.0)
    + reg.geom.depth.num_value;
end if;

```

Software routines, which manipulate the relative position of features, to satisfy set-up planning and NC code generation requirements have been included in the Machine Planning implementation, described in chapter 9, while experiments which have resulted in successfully machined components are reported in chapter 11.

### 7.3.2 Feature Orientations on a Component.

The setups in which features can be machined relate to their spindle axis directions, as identified by Joshi [73]. As with feature coordinates above, which are initially defined at the (0,0,0) point with respect to the global coordinate system, the spindle axes directions and fixture face normal directions, are also defined, initially, relative to the global coordinate system. When features are subsequently applied to a component definition they are likely to be rotated, such that these directions will be changed. Figure 7.2(c) shows a hole which has been applied to a component, such that its spindle axes lie in the plus and minus y directions. Hence the orienta-

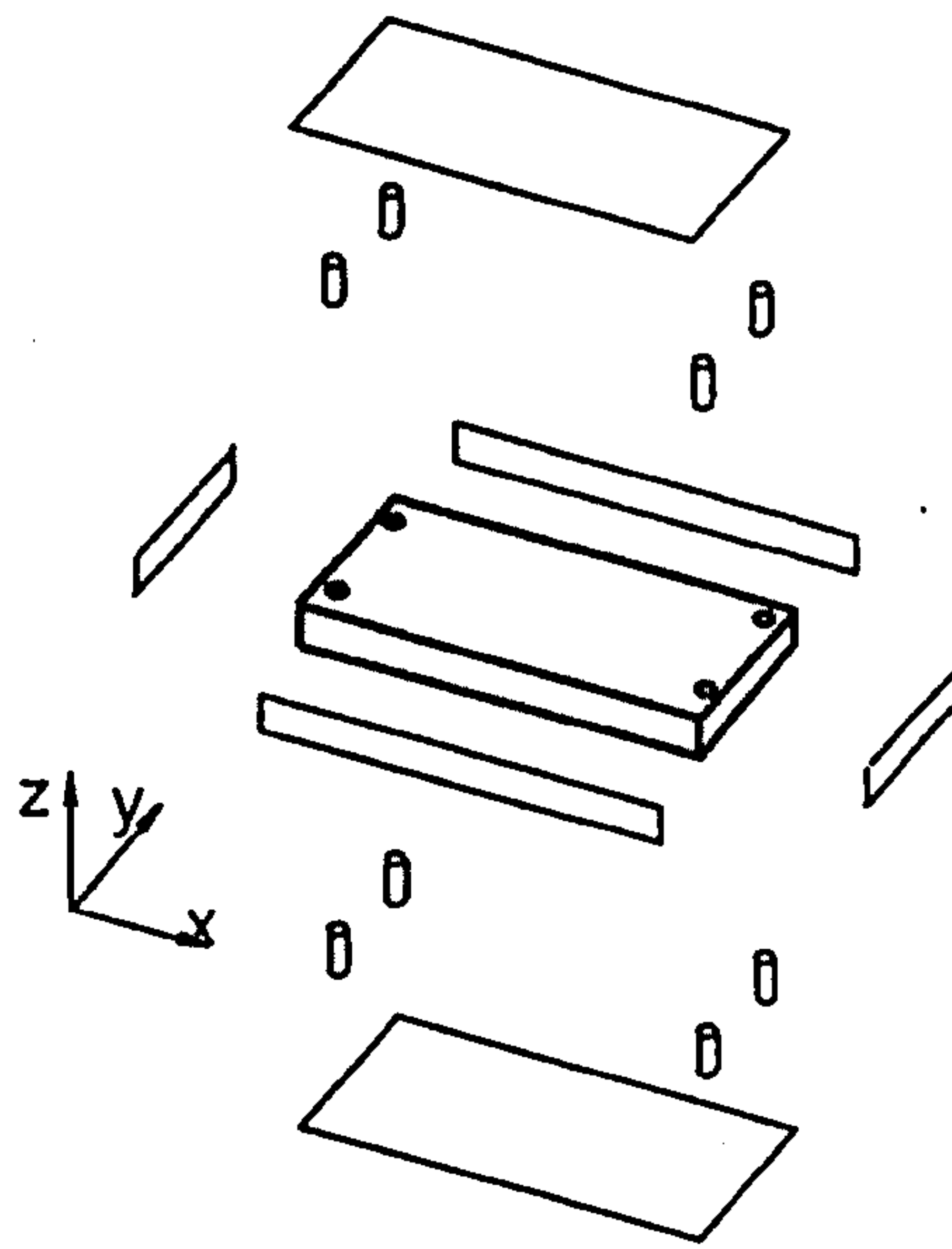


tion of features on the component must be identified.

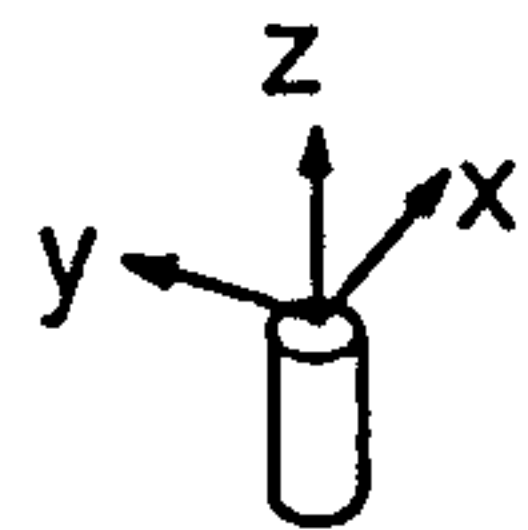
As in the case of feature position, the orientation of features has been tracked, by tracking the transformations applied to the features when they are applied to the component. This has been implemented as a part of the Identify Component Setups activity, described in chapter 9.3.

When features are grouped by spindle axis direction, duplicate features have been produced for each additional direction defined. A simple illustration of this is provided in figure 7.4 (a). The duplicate feature is defined with respect to a different setup from the initial feature. The reference axes for the duplicate feature have therefore been changed to suit the alternative setup direction, as illustrated in figure 7.4(b). This method is implemented in the setup sequence activity described in chapter 9.3.2.

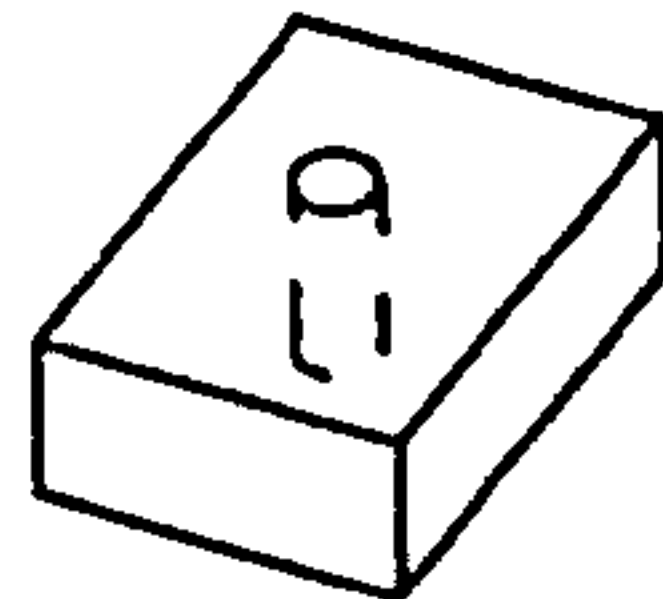
A feature's spindle axis direction defines the orientation, with respect to the machine tool spindle, from which the feature can be accessed. The Face feature is a special case of this and has been defined as having one spindle axis direction, opposite to the direction of the face normal. This has been done to ensure that the feature is machined using Facing operations, although the feature can, in fact, be accessed from all directions perpendicular to the face. In the use of the fixturing strategies, which will be described in chapter 9.2.3, there is a need to make use of these alternative directions of access to machine some Face features using End Milling operations. Use has been made of the alternative directions by re-orienting the spindle axis direction of the feature to align it with the setup direction being used, and converting the operation data to represent End Milling operations. The above requirement has been captured in the selection of fixturing strategies described in chapter 9.3.2.4



(a) Grouping Features by Spindle Axis Directions

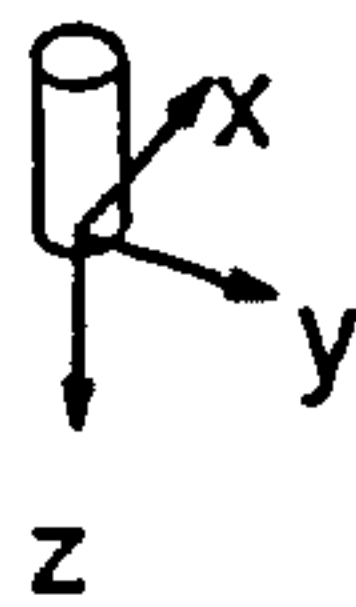


setup  
direction  
↓



component  
reference  
axes

through hole



↑  
setup  
direction

(b) Altering Reference Positions for Duplicate Features

Feature Manipulation  
Figure 7.4

The above methods of manipulation of features, to identify specific orientation data, have been implemented in the experimental Machine Planner defined in chapter 9 and have proven to be successful, as described in chapter 11, in generating plans and part programs which reflect the original component description, represented in the PDM.

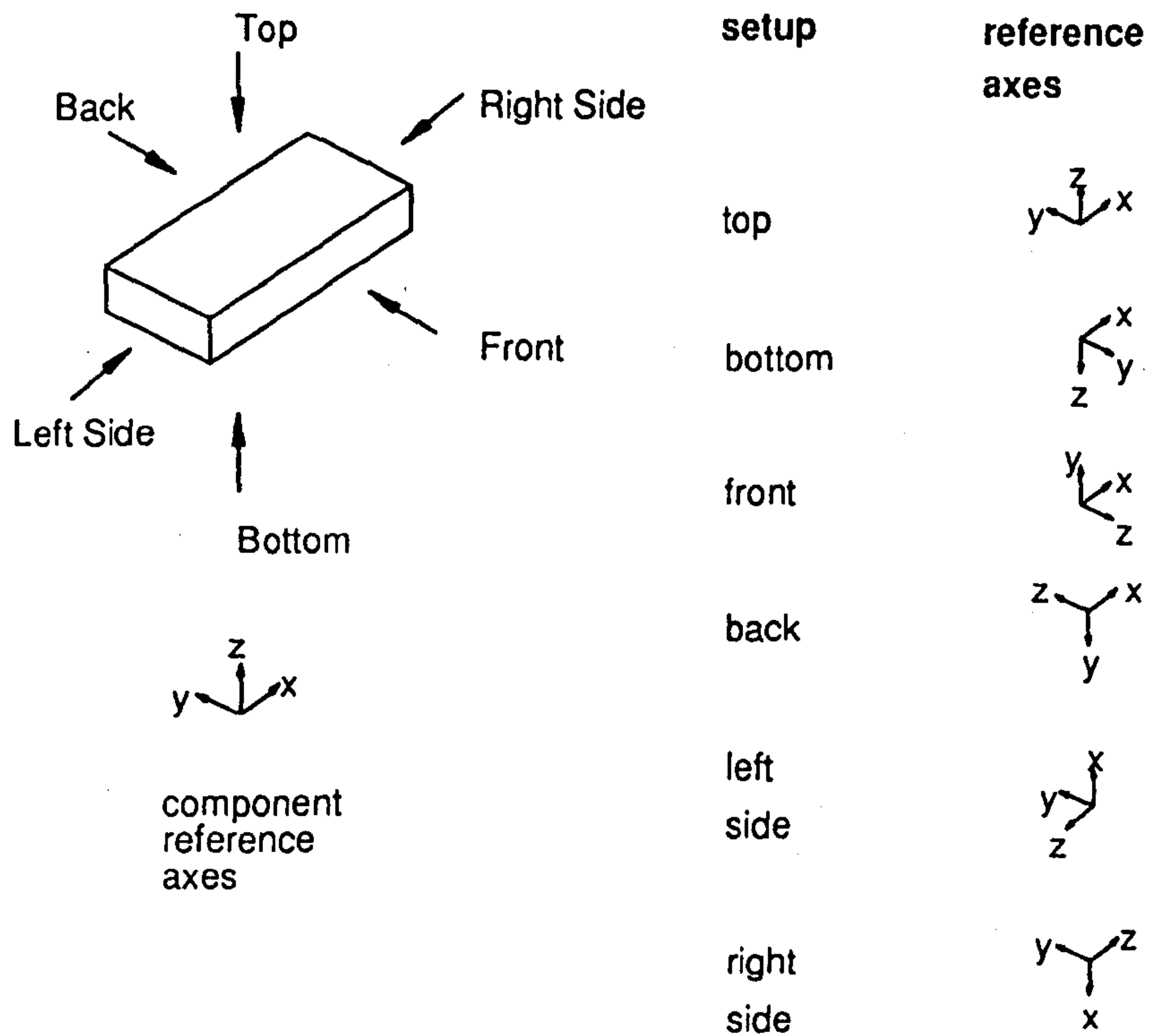
### **7.3.3 The Relative Position of Setups.**

In defining a setup sequence, from a fixturing strategy, as will be described in chapter 9.2.3, there is a need to have some positional awareness of the setups on the component, relative to each other. The approach taken in this research has been to describe the relationship between the six sides of the rectangular workpiece as top, bottom, left side, right side, back and front, as illustrated in figure 7.5(a). This figure also illustrates that setup axes have been altered to relate to the direction of approach to the setup. This data has been used, successfully, in the experimental Machine Planner, when identifying the preliminary setups to be used with a fixturing strategy

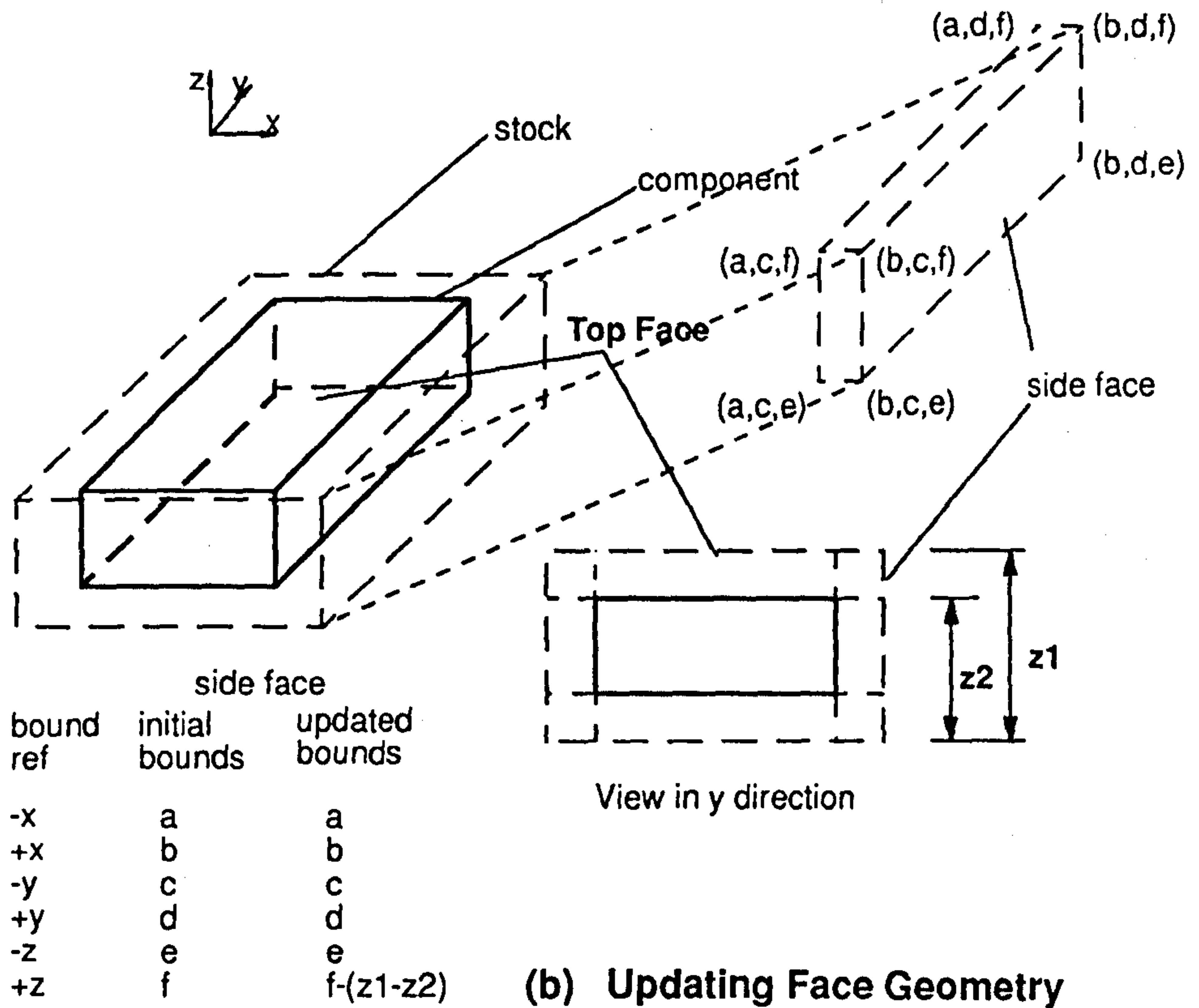
### **7.3.4 Updating Setup Geometry.**

Once a setup sequence has been identified, the pre and post machined geometry for each setup could be found by starting from the stock geometry and subtracting the feature geometries to be machined in each setup. This is a requirement which has not been implemented in the experimental software, but it is seen as being of little significance to the overall thrust of the research.





(a) The Setup Reference Axes



(b) Updating Face Geometry

Setup Data Manipulation  
Figure 7.5

Changes to the process data related to machining features, which come about through features overlapping across setups, will affect the resultant part programs which are produced. This type of overlap occurs with Face features, as discussed in chapter 6.2. Figure 7.5(b) illustrates a block which is to be machined on all faces. Until it is decided which one is to be machined first, each face must be considered to cover the complete extent of the stock. Once a sequence is chosen the areas of overlap should be addressed and the geometric data relating to the remaining machining updated. Hence, if the top face is machined, then the dimension  $z_1$  should be changed to dimension  $z_2$ . This can be achieved by representing the bounds of each face region in relation to its reference position. These bounds can be updated once the setup sequence is known. The need to update setup geometry was not identified until the author's research was well advanced. The data structure in the PDM, which can store the necessary data has been included in the feature *process data* sub-set of the PDM, described in chapter 6, but has not been proven through links to the experimental software.

The feature manipulation described up to this stage in the chapter has been achieved by linking to the CSG model of the Product and using the translation data associated with the features on the component. The following section argues how links to a second geometric description, the SDSM, allows feature interaction data to be identified.

## 7.4 THE METHOD USED TO IDENTIFY FEATURE INTERACTION DATA

This section investigates how links made, through the PDM to the Spatially Divided Solid Model (SDSM), can be used to generate necessary information on feature interactions, which are needed by a Machine Planning Application. The implemented links, included in the experimental Machine Planner are reported in chapter 9.3.2.4. These have been used, as described in chapter 11.3.4, to demonstrate that the integration of geometric Applications through the PDM has provided a basis for the identification of feature interaction data.

### 7.4.1 Geometric Queries on a Cell Decomposition Model.

The need to identify feature interaction data has been identified by Willis [149] and by Kanumury [76]. The author has explored geometric queries on the SDSM as a means of checking the geometry surrounding a feature. A cell's position, or address, in the SDSM can be found in relation to a position in the CSG model using queries provided with the SDSM. Each specific cell in the SDSM can provide data, which identifies whether the cell is empty, full or partial, as well as identifying the cell position in the representation as described in appendix 1. When a cell is partial, i.e. contains a face or faces, a pointer to the list of faces is provided from the cell. From this basic data specific queries can be built to provide answers to specific geometric questions such as:

Is the surface around this hole clear?

Is the surface under this clamp clear?

Is there a thin wall problem around this feature?



The following sections describe the methods explored to answer the above queries, while the experiments described in chapter 11.3.4 demonstrate that the interactions between a Machine Planner and geometric Applications enable such questions to be answered.

One problem with the use of cell decomposition models is that they can use a large amount of computer memory and analysis of such a representation can take significant amounts of time. During testing it was found that to create a cell decomposition model with a level six decomposition, i.e. a breakdown which subdivides the cells six times, would take a few minutes. A level seven decomposition would take around 10-15 minutes with a subsequent increase in the time for queries to be performed. Higher decomposition levels would have resulted in unacceptably high processing times on the computer equipment used, although advances in the processing capability of computers suggest that this will not be a serious limitation for future integrated systems.

Cell size is the limiting factor as far as the queries here are concerned. The cells had to be small enough to give a reasonably accurate answer, but as large as possible to minimise processing time. A cell size of two millimeters was chosen for the fixture related queries, as this was felt to be accurate enough for the type of queries being used. This resulted in decomposition levels of six for the Glacier test piece experiment described in chapter 11.4 and level seven for the bolster plate experiments described in chapter 11.3.4. Given the cell size, the level of decomposition was dependent on the size of component. This cell size would not be adequate for thin wall queries, unless all walls could be guaranteed to be a minimum thickness of three millimetres. This is due to the accuracy of the checks, as described in chapter 7.4.5.

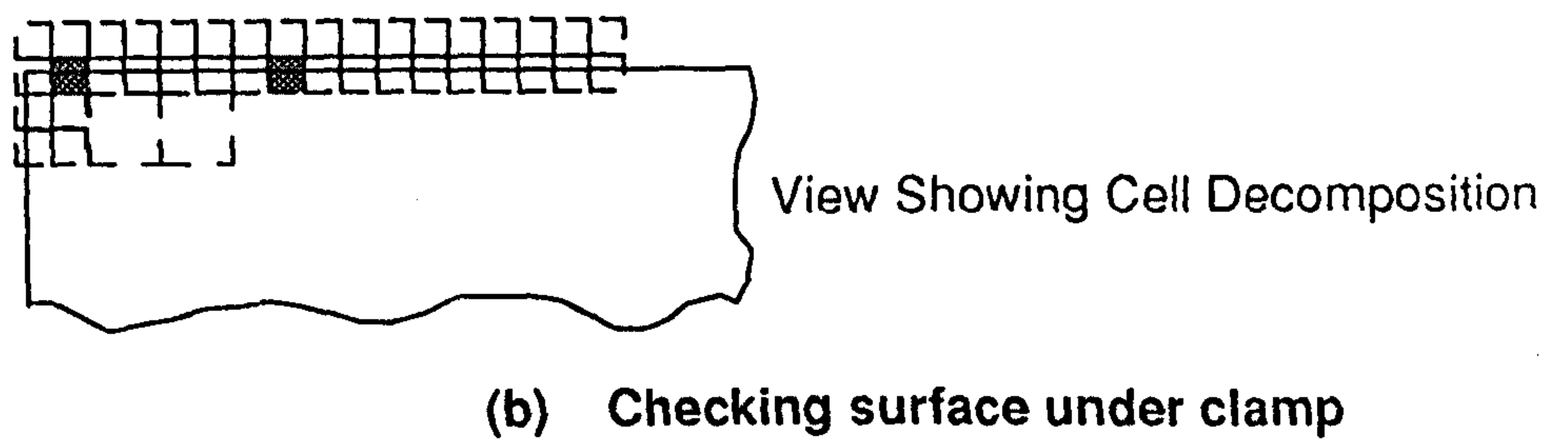
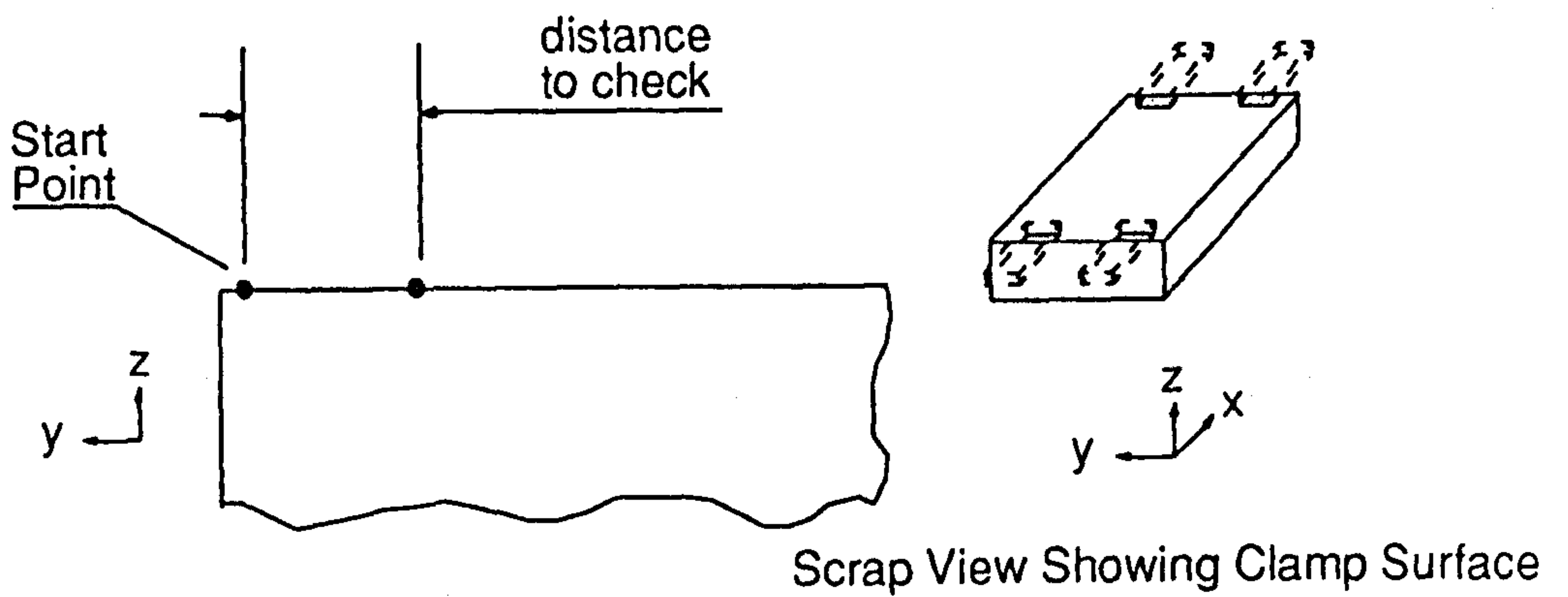
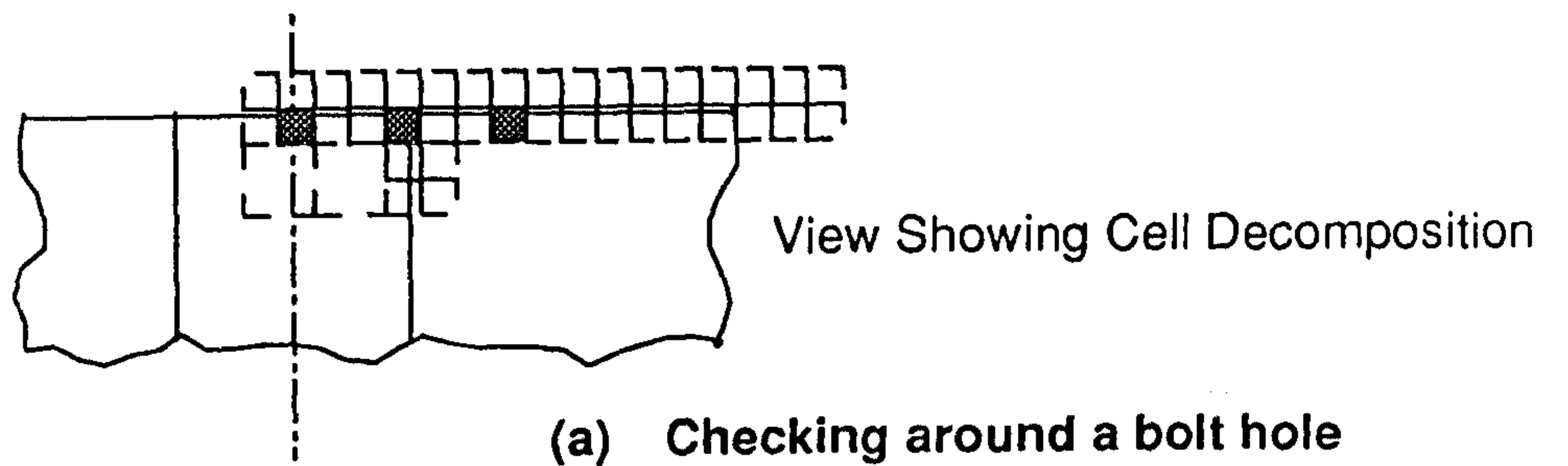
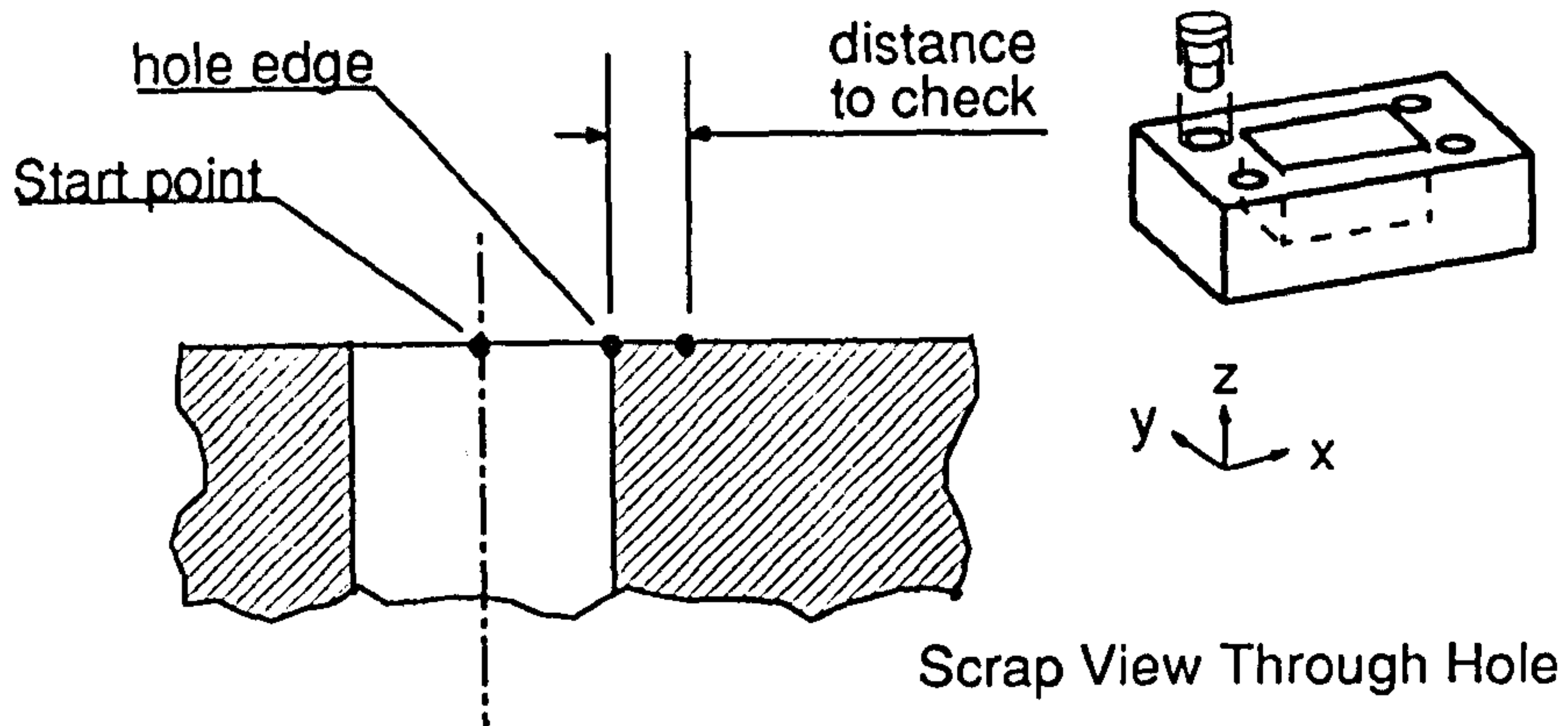
### 7.4.2 Checking the Surface Around a Hole.

To ensure that a hole is suitable for use in clamping requires that the cells around the hole surface are checked, to ensure that they contain only the face against which the bolt will clamp. The illustration in figure 7.6(a) shows the *distance to check* as the overlap of the clamp on the face. In this research the *distance to check* has been taken to be half the hole diameter as this represents the size of a bolt head.

Before the surface can be checked the hole edge must be found. This has been found by using the feature reference position, with respect to the global coordinate system, as a start point. This position has then been used to identify the cell in which the start point is contained using the ADA function:

```
function get_region_cell_address
-----
( part      : in part_params.part_data;
  reg      : in region_params.region_ptr)
return sdsd_cells.cell_address;
```

This function takes as input the cell decomposition representation, or SDSM object, for the part and the reference position of the region in question. It returns the cell address for the cell which contains the region reference position, i.e. the start point. The cell at this address will generally be empty, in the case of a hole, or will be partial if the cell contains the edge of the hole.



### Surface Checks Using SDSM

Figure 7.6



The edge of the hole has to be found next. This has been achieved by stepping from the start point, through adjacent cells, to the first partial cell. This contains the hole edge. The function used to achieve this is:

```
function next_valid_cell
-----
(rep      : in sdsd_maintenance . sdsd ;
in_cell  : in sdsd_cells.cell_address;
axis     : in axes.three_axes.three_axis;
pos      : in boolean) return
sdsd_cells.cell_address ;
```

This function takes as input the SDSM object in question, *rep*, and the cell address from which to start, *in\_cell*. In addition the direction in which to check the cells needs to be specified. This is provided by the *axis* and *pos* values, where *axis* can be x, y, or z with the direction being positive when *pos* is true and negative when *pos* is false. The first partial cell in the direction specified is returned.

Once this first partial cell has been found the surface to be checked has been reached and the function which is to carry out the check on the surface can be called. The function used to check the surface is:

```
function clear_dist
-----
(rep   : in sdsd_maintenance . sdsd ;
in_cell : in sdsd_cells.cell_address;
dist   : in gmp.real;
axis   : in axes.three_axes.three_axis;
pos    : in boolean)
return boolean ;
```

This function takes as input the SDSM object, *rep*, and the cell containing the hole edge, *in\_cell*. The distance to be checked is needed, *dist*, as is the direction in which to check, *axis* and *pos*. Each adjacent cell is then checked to identify if any face, other than the clamp face, is contained in the cell. If there is another face this is taken to be a problem and the function returns a false value. If the "distance to be checked" is covered successfully then a true value is returned.

The above functions have then been used to check around a feature using the function *clear\_around\_region*:

```
function clear_around_region
-----
(rep          : in sds_maintenance . sds_m ;
 mid_cell     : in sds_m_cells.cell_address;
 requ_dist    : in gmp.real;
 spindle_axis : in axes.three_axes.three_axis)
return boolean ;
```

This function takes as input the SDSM object, *rep*, and the start point in the hole, *mid\_cell*. Also input are the distance to check, *requ\_dist*, and the relevant *spindle axis* for the setup. The function then uses the *next\_valid\_cell* and the *clear\_dist* functions, described above, in each of the orthogonal directions perpendicular to the spindle axis direction. If the *clear\_dist* function returns true in each of the directions checked then the function returns true, otherwise it returns false and the hole is not passed as clear.

The *clear\_around\_region* function is the main function which confirms whether, or not, a hole is suitable for clamping, in terms of interactions with other features. The experiments performed using this function to explore fixturing strategy selection are described in chapter 11.3.4, and show that the method employed is appropriate for capturing feature interactions of this nature. A partial listing of the ADA code for the *clear\_around\_region* function is provided in appendix 5.4

### 7.4.3 Checking the Surface Under a Clamp.

To ensure that a surface is suitable for use in clamping, requires that the cells on the surface, under the clamp position, are checked, to ensure they contain only the face against which the clamp will rest. The illustration in figure 7.6(b) shows the clamp width as the *distance to check*.

Before the surface can be checked a point on the surface must be identified, to provide a basis for starting the check. This has, initially, been assumed to be five millimetres in from the component edge. However, if the cell at this point is unsuitable then the start point must be moved. This has been achieved by using the function below which checks the start point. If it is found to be unsuitable then it moves along the edge of the component until a suitable start point is found.

```

function check_start_pt_clamps
-----
(rep      : sds_maintenance . sds_m ;
in_cell  : sds_m_cells.cell_address;
axis     : axes.three_axes.three_axis;
pos      : boolean)
return   sds_m_cells.cell_address ;

```

The *check\_start\_pt\_clamps* function takes as input the SDSM object, *rep*, and the cell containing the proposed start point, *in\_cell*. The direction in which to check for and alternative start point, *axis* and *pos* is also input. The cell address for the start point is returned by the function. Once this has been found the face under the clamp can be checked using the procedure:



```

procedure dist_to_face
    -----
(rep      : in sdsd_maintenance . sdsd ;
in_cell  : in sdsd_cells.cell_address;
axis     : in axes.three_axes.three_axis;
pos      : in boolean;
dist_to_check : in gmp.real;
dist_ok  : out boolean;
dist     : out gmp.real;
out_cell : out sdsd_cells.cell_address) ;

```

This function takes as input the SDSM object, *rep*, and the cell containing the start point for the check, *in\_cell*. The distance to be checked is needed, *dist\_to\_check*, as is the direction in which to check, *axis* and *pos*.

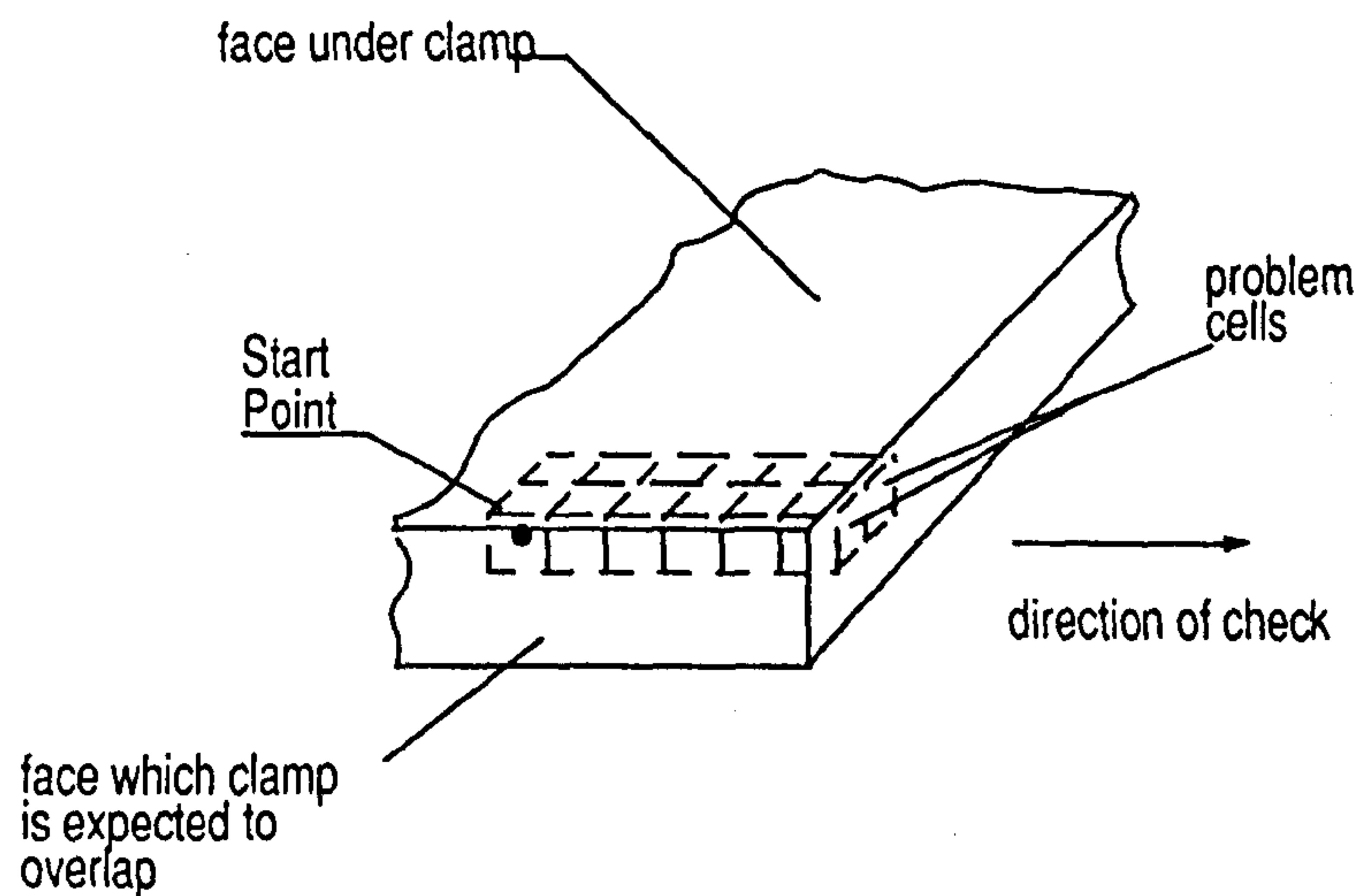
The procedure then steps across the face, checking each face list in each partial cell using the *face\_normal\_in\_view\_dirn* function shown below. Assuming the surface is clear, over the distance required, the *dist\_ok* returns a true value. If this is not the case then the distance where the problem occurred, *dist*, and the problem cell, *out\_cell*, are identified.

```

function face_normal_in_view_dirn
    -----
(faces    : in face_lists.list;
axis     : in axes.three_axes.three_axis)
return boolean;

```

*face\_normal\_in\_view\_dirn* checks each face found in the cell. The *faces* to be checked are input as is the view direction, i.e. the *axis* in which the check is taking place. If a non-planar face is found then it is assumed that an alternative feature has been found and a problem with clamping is assumed. If a planar face is found it may belong to the surface being checked, or the side face of the component, or be a problem face as illustrated in figure 7.7. Hence the normal to each planar face is checked. If it lies in the same direction as the check which is being made it is assumed to imply a problem for clamping. When a problem for clamping is



### Checking Planar Faces in a Cell

Figure 7.7

found then the function returns a true value.

The functions argued here have been incorporated in the Machine Planner implementation described in chapter 9.3.2.4, and used in the experimental exploration of the Closely Coupled Machine Planner, as will be described in chapter 11.3.4. In this way, the successful use of the SDSM representation to provide feature interaction queries will be demonstrated. The achievement of this major requirement for a Machine Planning Application demonstrates one of the values of using an integrated environment to link appropriate Application packages together.

#### 7.4.4 Checking for Thin Walls.

The combination of the *next\_valid\_cell* function and the *dist\_to\_face* procedure can be used to provide the basis for a procedure to check for thin wall sections around a feature using a procedure:

```

procedure check_thin_walls
-----
(rep          : in sds_maintenance . sds_m ;
mid_cell     : in sds_m_cells.cell_address;
requ_dist    : in gmp.real;
axis         : in axes.three_axes.three_axis;
pos          : in boolean;
thin_wall    : out boolean;
thickness    : out gmp.real;
prob_cell    : out sds_m_cells.cell_address);

```

Initially the edge of the feature being checked can be found using the *next\_valid\_cell* function. Then the required distance to be checked can be input to the *dist\_to\_face* procedure which will either identify that there is no thin wall, or provide the wall thickness and the cell at which the problem occurs. This procedure provides a start point in checking for thin walls but only provides a narrow view through one set of cells. The procedure could be readily extended in future work to provide greater coverage of the volume surrounding a feature. To ensure sufficient accuracy in this procedure, a smaller cell size than that used in fixturing interaction checks would be advisable, as described in the next section. The *check\_thin\_walls* procedure has not been implemented in the experimental software.

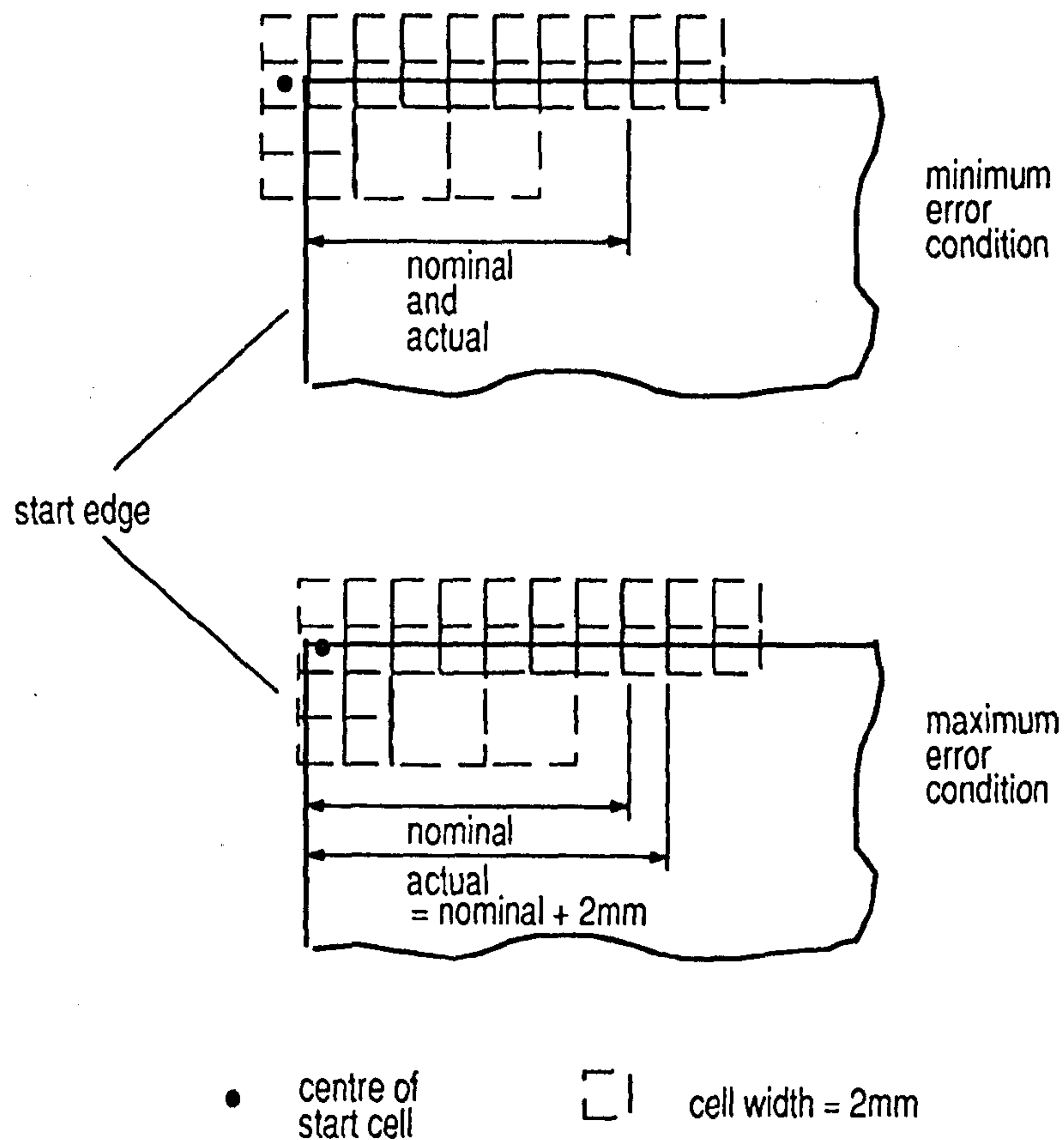
#### 7.4.5 Accuracy of Checks.

The accuracy of the feature interaction checks, using the SDSM, is dependent on the cell size used and the surface coverage of the algorithm.

The actual distance checked in the queries will generally be greater than the nominal *distance to check*, with a maximum error of plus one cell width. In the case of a 2mm cell size the distance is measured from the cell centre at the hole edge, in 2 millimetre increments until the *distance to check* has been exceeded. Hence the variation from the nominal distance can be zero to a maximum of 2 millimetres as



shown in figure 7.8, dependent on the position of the cells relative to the start



### Accuracy of distance checks

Figure 7.8

point. This has been assumed to be adequate in the case of the fixture based queries which have been implemented, but a significantly smaller cell size would be needed to check thin wall problems. A cell size of 0.5mm would still check a surface 0.5mm beyond the specified *distance to check*.

The software implementation of the geometric queries has used the orthogonal axes, perpendicular to the spindle axis direction, as the directions in which to perform geometric queries. The use of this approach has illustrated the ability to link to the SDSM representation to check feature geometric interactions, although more complex algorithms could provide more comprehensive queries. The importance of the use of geometric queries, to the author's research, is that the implemented

Closely Coupled Machine Planner, described in chapter 9 and 10, and demonstrated in experiments described in chapter 11, show that queries on an SDSM can provide a solution to problems of feature geometric interactions. It also shows that the integration of geometric Applications through the Product Model environment can be achieved, which provides the potential for more powerful links between Applications in the future.

Another important aspect of feature interaction is that of dimensional interactions, as discussed in chapter 6.2.3. The problem of linking dimensional data through a Product Data Model, to a Closely Coupled Machine Planner is discussed in the following chapter.

## CHAPTER 8

### THE DIMENSIONAL FACTOR IN A CLOSELY COUPLED MACHINE PLANNER

#### 8.1 INTRODUCTION.

This chapter discusses the challenge and opportunities which can be offered by linking dimensional data to a Closely Coupled Machine Planner. The dimensional representations available in the PDM, i.e. parameter values and the Relationship Graph, are described. The method by which dimensional data has been linked to the Machine Planner is explained, which is then used in the realisation of the experimental Machine Planning Application described in chapter 10.

The limitations of the available methods to provide an appropriate source of dimensional data for Machine Planning is discussed and an argument for methods required in the future is made. A method of linking the Relationship Graph to features is argued, to provide a source of inter-feature dimensional data for Machine Planning. This provides an initial source of dimensional data for planning, while links to tolerance analysis Applications, made possible by the integrated environment, is further argued, as a future requirement to confirm the suitability of generated plans.



## 8.2 DIMENSIONAL AND MACHINE PLANNER INTERACTIONS

A stand alone Machine Planner requires dimensional data to function effectively. Dimensional relationships within features influence the cutter which should be used and the processing parameters which should be used in a part program. Dimensional relationships between features can influence the need to machine two operations with a common tool, a problem identified by Carlier [24] and inter-feature dimensional relationships can also influence the need to machine two operations in the same setup, as investigated by Boerma [17].

An integrated Machine Planner opens up the possibility of linking to tolerance analysis Applications, providing different views of the dimensional representation of the component. A designed component will have a dimensional representation which reflects the functional requirements of the component. However, as has already been argued in chapter 6.2.3, a Machine Planner requires dimensional relationships which reflect the machining features on the component and their interrelationships. This is a different view from the functional design view. Furthermore, once setup and operation plans have been generated, a further dimensional representation is generated, which is based around the fixture datums in each setup. Hence there is a need in future systems to capture and assess these different views of the same component. This aspect of interaction can be accommodated by links between a Machine Planner and tolerance analysis packages, as argued in section 8.4.2.

Capturing dimensional relationships in a PDM also opens up the ability to share such data across a range of Manufacturing Applications, ensuring current versions of data are always used. An example of linking dimensional data in this way can be found in the MCG Applications of the ISS project, illustrated in figure 4.5. This shows how the data from the Relationship Graph is required by all three Applica-

tions and can be accessed from the central Product Data Model. Section 8.3.2 describes the problem with linking a Machine Planner directly to the Relationship Graph, while section 8.4.1 argues a method by which Relationship Graph and feature data can be brought together to identify inter-feature dimensional data.

### **8.3 DIMENSIONAL REPRESENTATIONS IN THE PRODUCT DATA MODEL.**

This section describes the dimensional representations available in the PDM and their limitations for use in Machine Planning.

#### **8.3.1 Parameter Values.**

Parameters in the Structure Editor can be used to define tolerance values of the form:

$$\text{tolerance value} = \text{nominal value} \pm \text{tolerance}$$

These have been used in the definition of pre-defined machining features as described in chapter 10.4.1 and have been used to represent key component dimensions as described in chapter 10.4.3. This has proved to be an inadequate means of representation, however, as parameter values cannot be used to represent inter-feature dimensions. There is no apparent representation of data, inter-linking features, to which parameters can be applied. Parameters have been linked to the geometric description of the component which in turn have been linked to the Relationship Graph(RG) method of representing dimensions and tolerances, described in the next section. This provides a aid to the identification of inter-feature tolerances as argued in section 8.4.1

Parameters, related to features, have also been used to define the surface roughness requirement for a feature, as described in chapter 10.4.1. However, this type of parameter is not linked to geometry and therefore cannot, in turn, be linked into the RG.

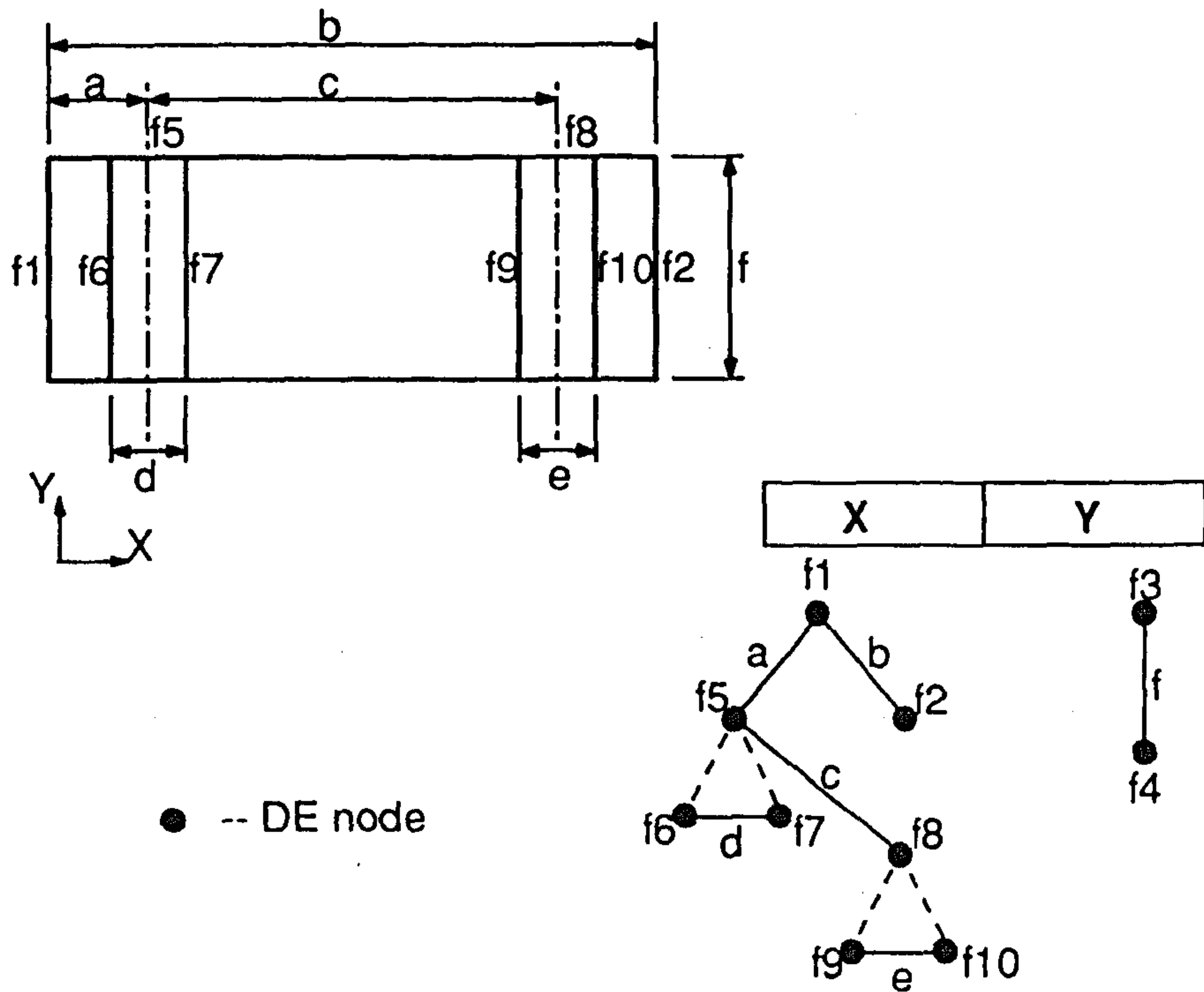
### 8.3.2 The Relationship Graph

The Relationship Graph (RG) method of representing dimensions and tolerances, and linking them to a geometric description has been researched by Wickens throughout the ISS programme and a full description of the technique is provided in [148]. This has provided a mechanism for the representation of dimensional and geometric tolerances and surface texture.

The Relationship Graph describes sets of Dimension Entities (DE) and the Constraints which apply to them. A Dimension Entity is a group of geometric elements, commonly a face or a centre line. A Constraint represents dimensional data covering dimensional tolerances, geometric tolerances or surface texture. In his work, Wickens has shown how the RG can be used to represent dimensional data and how this can be linked to geometric descriptions. Relationships are defined for each of the x,y and z directions relating to a component model. The RG starts from a coordinate relationship which consists of a mutually perpendicular set of Dimension Entities. This defines the x,y and z legs of the RG. Figure 8.1 provides a 2 Dimensional representation of a Relationship Graph where f1 -> f10 represents the DE nodes and a -> f represents the constraints between them.

There is no association from the RG to the feature representation in the PDM. The RG provides only a link to geometry through the DE nodes. In addition, although the RG provides dimensional relationships in the x,y, and z directions there is no direct association from this to the direction of approach which could be used to ac-





**A 2 dimensional Relationship Graph Example  
Figure 8.1**

cess a particular dimensioned face. For example, consider the dimension 'd' in figure 8.1. Traversing the RG provides no information to tell whether the associated DE's, f6 and f7, are on the top or the underside of the component. The next section argues a method which can be used to identify inter-feature tolerances, assuming the dimensional relationships represented in the RG provide a view of dimensions which reflects a feature based tolerance scheme.

#### 8.4 INTER-FEATURE CONSTRAINT IDENTIFICATION.

##### 8.4.1 A Method of Linking the Relationship Graph to Features.

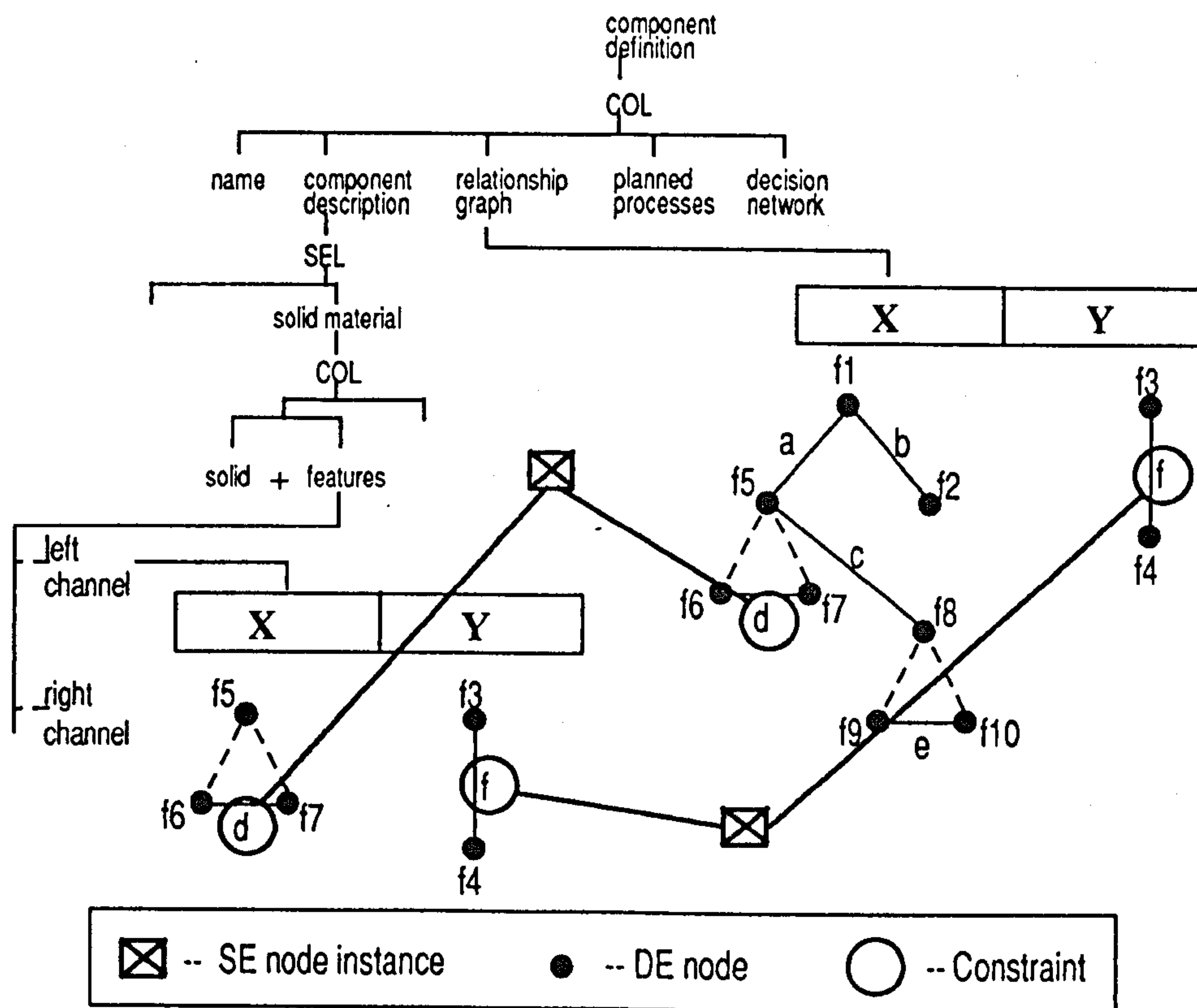
Although the Relationship Graph has been linked to geometry, there is no association from the RG to the feature representation and, therefore, there is no direct method by which a dimension can be identified as being an inter-feature constraint

or being contained within a feature. Although geometry would appear to be a common link between features and the RG, this does not provide a straight forward link, as dimensional constraints are linked to Dimensioned Entities, which may be centre lines, or groups of faces, which do not link back readily to the feature description in the ISS.

Geometric tolerances and surface texture will need to be linked to features, or between features, through an analysis of the geometry. There are, however, two common links between the RG and feature descriptions which enable inter-feature dimensional constraints to be identified. These are parameter values and the Structure Editor "node instance". When generating a Relationship Graph all parameters which have been used in the component definition are evaluated and tolerance values included in the RG. Hence parameters used in feature definitions will be carried through to the RG.

All nodes in the PDM are built using the SE which provides that each node is of a type called a "node instance" which can then have, below it, data relating to whatever data structure was defined in the PDM. This "node instance" provides pointers into the PDM which allows software routines to access the particular part of the data structure for which the "node instance" is known. Where data is common, but appears in more than one part of the PDM, then it is represented only once, and software links are provided to the areas in the PDM where the data is to be found. It is therefore possible, where two parts of the data structure share the same data, to identify this by comparing node instances as these will also be shared.

Figure 8.2 provides an illustration of the component level RG and the feature level RG, highlighting common node instances. Each feature's RG can be traversed and the node instance for each of the feature related dimensional constraints identified. With this information the component level RG can then be traversed and the node



### Comparing constraints using the SE Node Instance

Figure 8.2

instance of each component related dimensional constraint compared against the node instances related to features. If a match is found then the component constraint is also a feature based constraint. If no match is found then it is an inter-feature constraint. The next DE in the list will identify the feature to which it is linked. Each of the x,y and z DE's can be checked in this way to provide a complete representation of feature and inter-feature tolerances.

In addition to identifying inter-feature constraints this procedure can also be used to identify the related direction of approach of the constraint, which is not held with the RG, as mentioned at the end of the previous section. The feature spindle axis directions can be used to identify the views from which dimensions are visible.



The nature of the ISS programme has been such that the work of researchers has been progressing in parallel. Because of this, the Relationship Graph, linked to the PDM, has only recently been available. Attempts, to date, to link this into the Machine Planner have raised Relationship Graph software bugs, which have not, as yet, been resolved.

#### 8.4.2 Tolerance Analysis Requirements in Setup Planning

The dimensional relationships on a component description, as it comes from a designer, reflect the functional requirements of the design, and may not take manufacturing methods into consideration in any detailed way. However, a Machine Planner needs a representation in terms of feature and inter-feature relationships. There is a need, therefore, for some analysis of the tolerance relationships to take place, in order to generate a feature based relationship, as illustrated in figure 8.3.

Feature and inter-feature constraints can be used in making planning decisions. Research, such as that of Boerma [17] assessing inter-feature constraints, could be applied to eliminate fixture strategies which are unable to meet the tolerance requirements. For example the necessary grouping of side faces into a single setup to achieve squareness constraints could eliminate the use of the Side Clamping Strategy.

Having generated setup and operation plans, new sets of dimensional relationships will exist, which relate to the manufacturing of the component. Each setup will have a datum. Each operation in a setup will be related to that datum and will have a tolerance associated to it which reflects the accuracy which can be achieved in manufacture. Similarly tolerances can be related between features in different set-



The value of working within an integrated environment, is that tolerance analysis Applications could be added into the system, providing a basis for a more comprehensive design and manufacture Applications. This is an area of major importance for future integrated systems which should be further investigated.



## CHAPTER 9

### THE EXPERIMENTAL MACHINE PLANNING APPLICATION

#### 9.1 INTRODUCTION.

This chapter explains and defines the Machine Planning functions, strategies and rules which have been implemented in order to explore the central hypothesis of the thesis, that a Machine Planner can function effectively as a Closely Coupled Application in a Product Modelling Environment. This work links directly on to chapter 10, which explains how the Product Data Model has been populated, and linked to the Application. Thereafter the success of the approach taken is demonstrated through the experiments reported in chapter 11.

The Machine Planning implementation has provided a sufficient basis for the exploration of Machine Planning as a highly automated subset of an integrated design and manufacture system. The implementation is derived from activities MP1, MP3 and MP4 of the Machine Planning IDEF0 model documented in appendix 3. Elements of the ADA code used in the implementation, are reproduced in appendix 5 and a guide to the use of the software is provided in appendix 6.

## **9.2 THE SCOPE OF THE MACHINE PLANNER IMPLEMENTATION**

### **9.2.1 Planning Functions to Explore the Product Model Environment**

Setup, operation and NC planning functions have been implemented, such that an experimental exploration of the Product Model environment, explained in chapter 4, could take place. The implementation has been aimed at the following:

1. To demonstrate that the proposed Product Data Model specification, which capture component and feature level data as argued in chapters 5 and 6, provides a comprehensive source of geometric, dimensional and planning data which is appropriate for the Machine Planning Application.
2. To show that the Integration of the Machine Planning Application can be achieved.
3. To demonstrate the viability of the integration of the Manufacturing Code Generation Applications of Machine Planning, Inspection Planning and Manufacturing Data Analysis through the Product Model environment.
4. To explore the links to the Geometric Representations described in chapter 4.4.1 and the Relationship Graph [148], through the Product Model environment, which can enhance the capability of a Closely Coupled Machine Planning Application.

### **9.2.2 The Implemented Planning Functions**

The overall aim of a Machine Planner is to plan how a component, or components, should be machined on a particular machine type and to generate appropriate machining plans and part programs. There are a number of views as to how planning functions should interact. Chan [25] proposes that setup planning should be-

come a subset of process planning, with operation planning becoming a downstream activity. CAM-I [47] have produced an IDEF0 model of process planning, with setup and operation planning as key activities, while recognising the need for the exploration of integration between operation and NC planning [113]. The view taken in the author's research of how MP functions should interact is provided in the IDEF0 model of Machine Planning in appendix 3. This has been produced following consideration of the Data requirements and outputs of the various functions, including the interactions between setup, operation and NC planning and between pre-defined and recognised features.

The implemented planning functions consist of the activities, using IDEF0 terminology, of *Identify Component Setups*, *Sequence Operations* and *Generate Machine Controller Code*. These are activities MP1, MP3 and MP4 of the Machine Planning IDEF0 model and are illustrated in figure 9.1(a). The break down of *Identify Component Setups*, *Sequence Operations* and *Generate Machine Controller Code* into sub-activities, and their software implementation, is explained in the following sections. Activity MP2, *Identify Machine Batch Setups*, is concerned with multiple component setups, and was not pursued as the author's work was constrained to consider only single components.

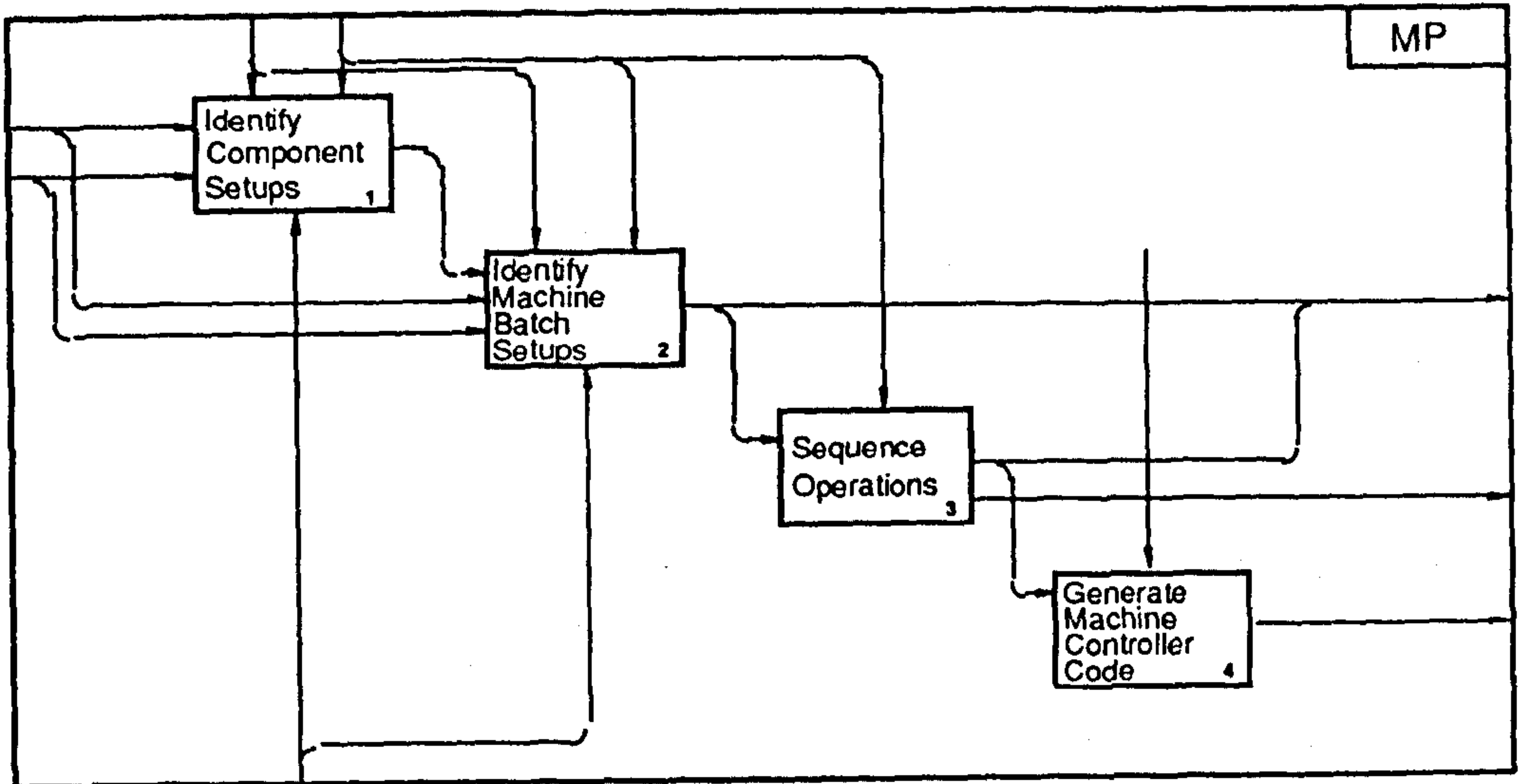
The sections explaining the software implementation follows the "top down" approach used in IDEF0 modelling. The actual implementation of the software, however, has followed a "bottom up" approach. This has enabled the lowest levels of the PDM, the Operation Data sub-set, to be explored experimentally as describe in chapter 11, before extending the exploration to higher levels of feature data, and then to component data and feature interactions.



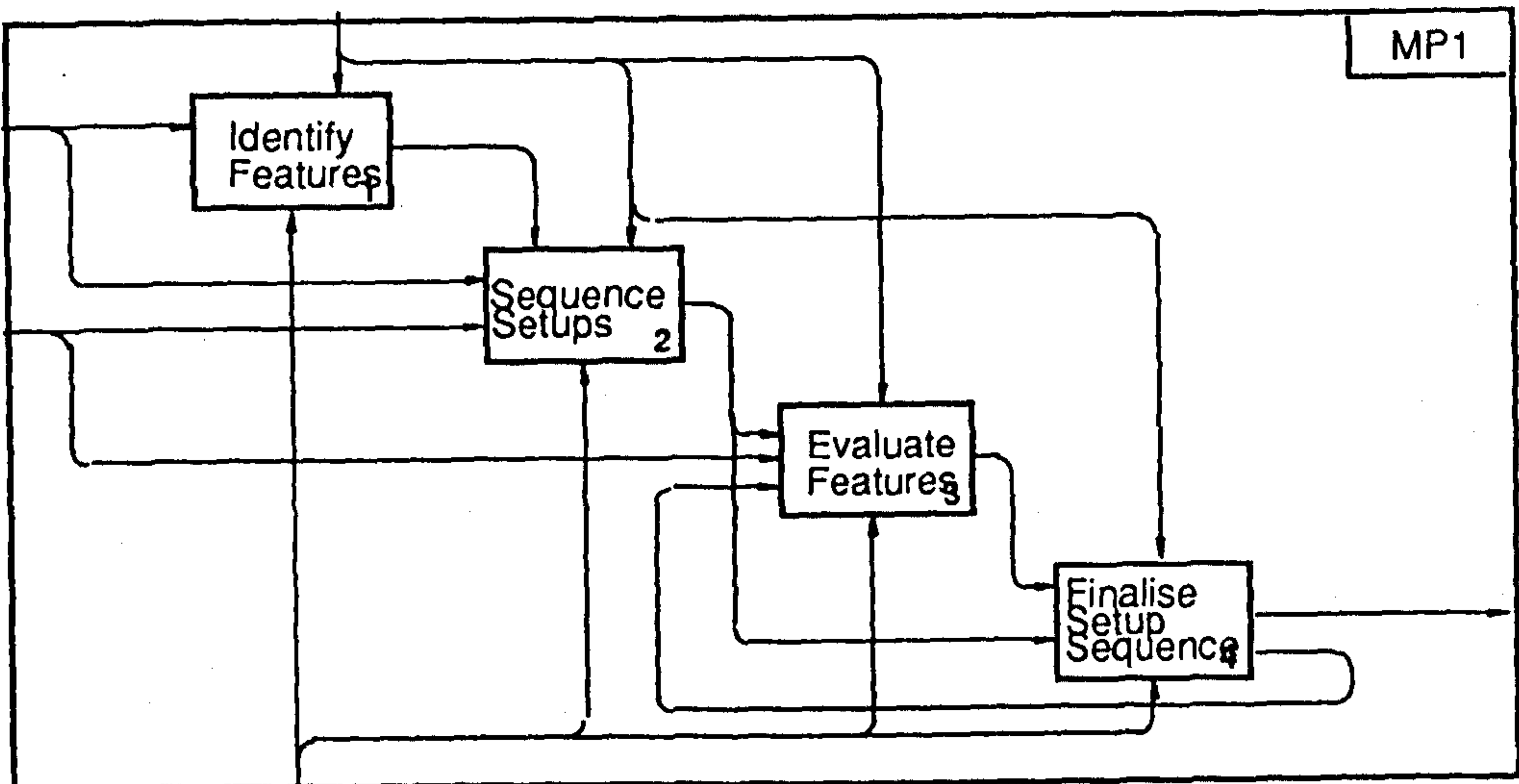
### 9.2.3 Fixturing Strategies and Machining Rules.

Fixturing strategies define ways in which different work-holding devices, on a particular machine tool type, can be used to fixture a component. There has been little exploration in the research community into the representation of such strategies in computerised setup planning. Willis [149], in considering different setup strategies, identified the need to consider the global geometry of the component and geometric constraints on the machining of the part, while Kanumury [76] used a strategy for the use of a vice when exploring fixture planning.

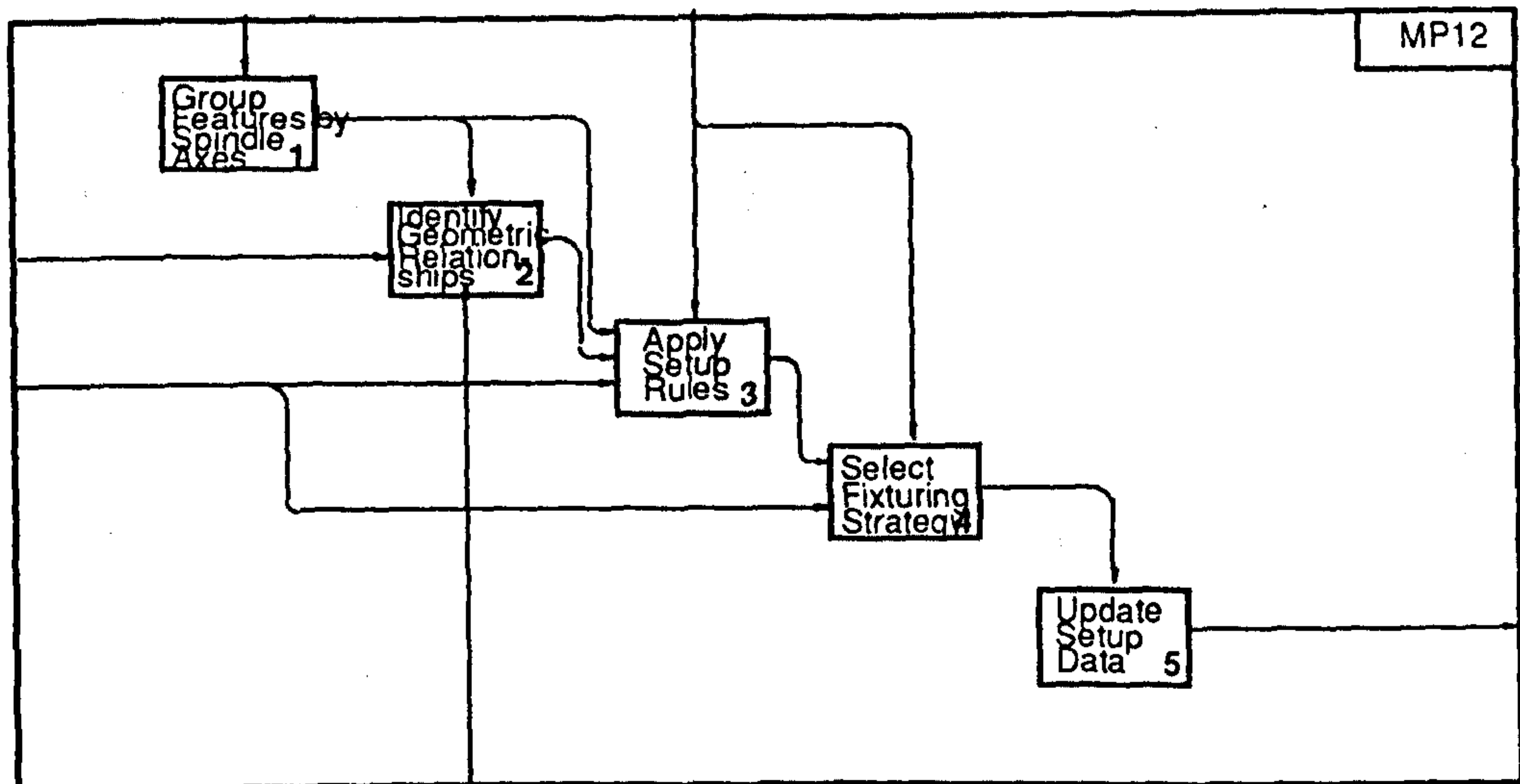
The important point coming from their work, as far as the author's research is concerned, is that component geometry can provide constraints on the identification of an appropriate strategy. As such there was a need to represent alternative ways in which components could be fixtured on the machine table, to explore how component data from the Product Model could be used to influence strategy selection. To provide a basis for this exploration, three simple fixturing strategies were implemented. These were Side Clamping, Through Clamping and Down Clamping as illustrated in figure 9.2. These, respectively, provided a mechanism whereby a setup sequence can be generated on the basis of the use of a vice, bolts or clamps. They also provide a means of capturing appropriate datum points for each of the strategies, as illustrated in the figure. The use geometric queries in strategy selection has been argued in chapter 7.4. These, too, are incorporated within the relevant fixturing strategy. An example of the software code, representing the Through Clamping Strategy, is provided in appendix 5; This appendix also provides a listing of the "clear around feature" query, defined in chapter 7.4.2.



(a) Machine Planning



(b) Identify Component Setups



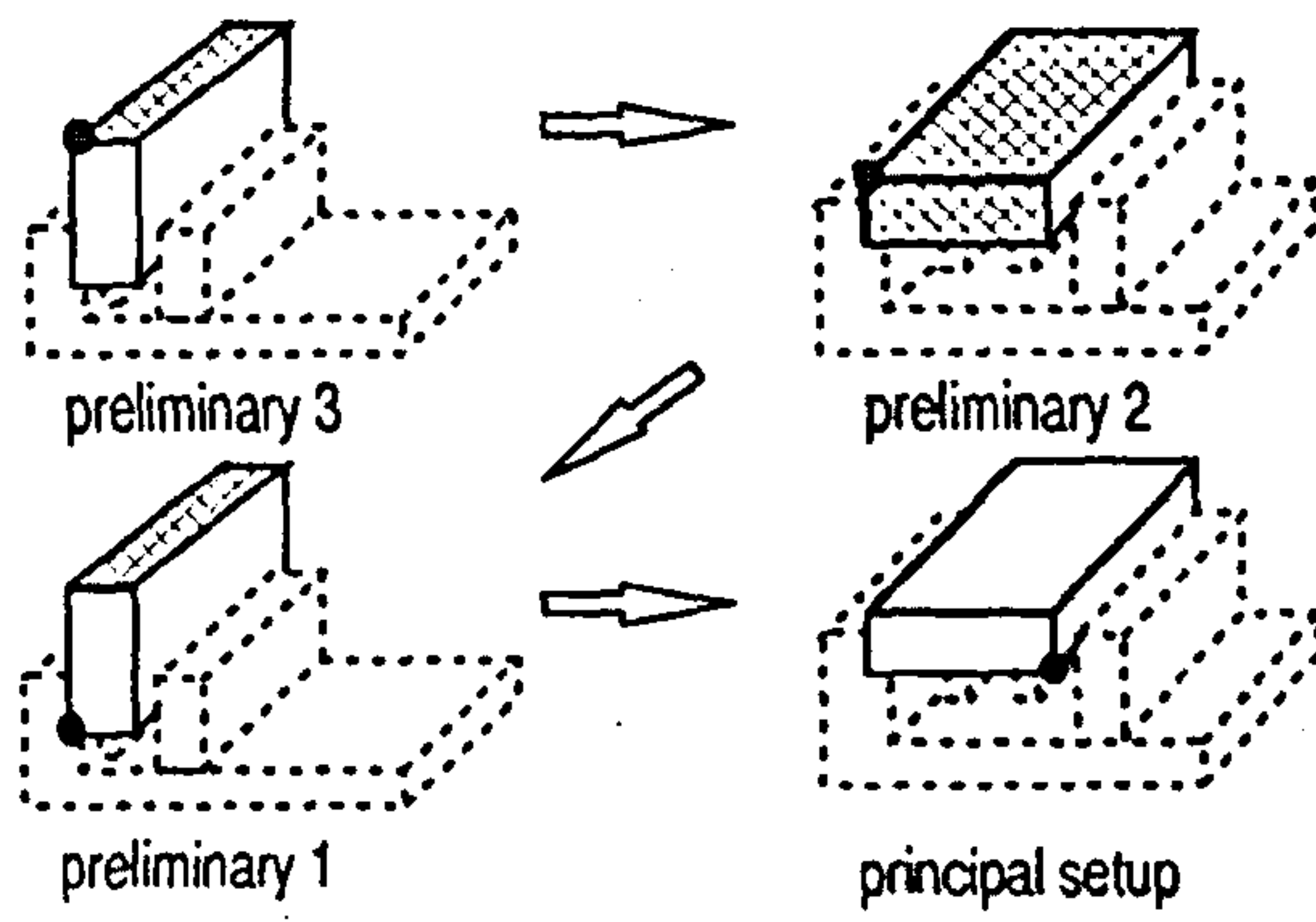
(c) Sequence Setups

Figure 9.1

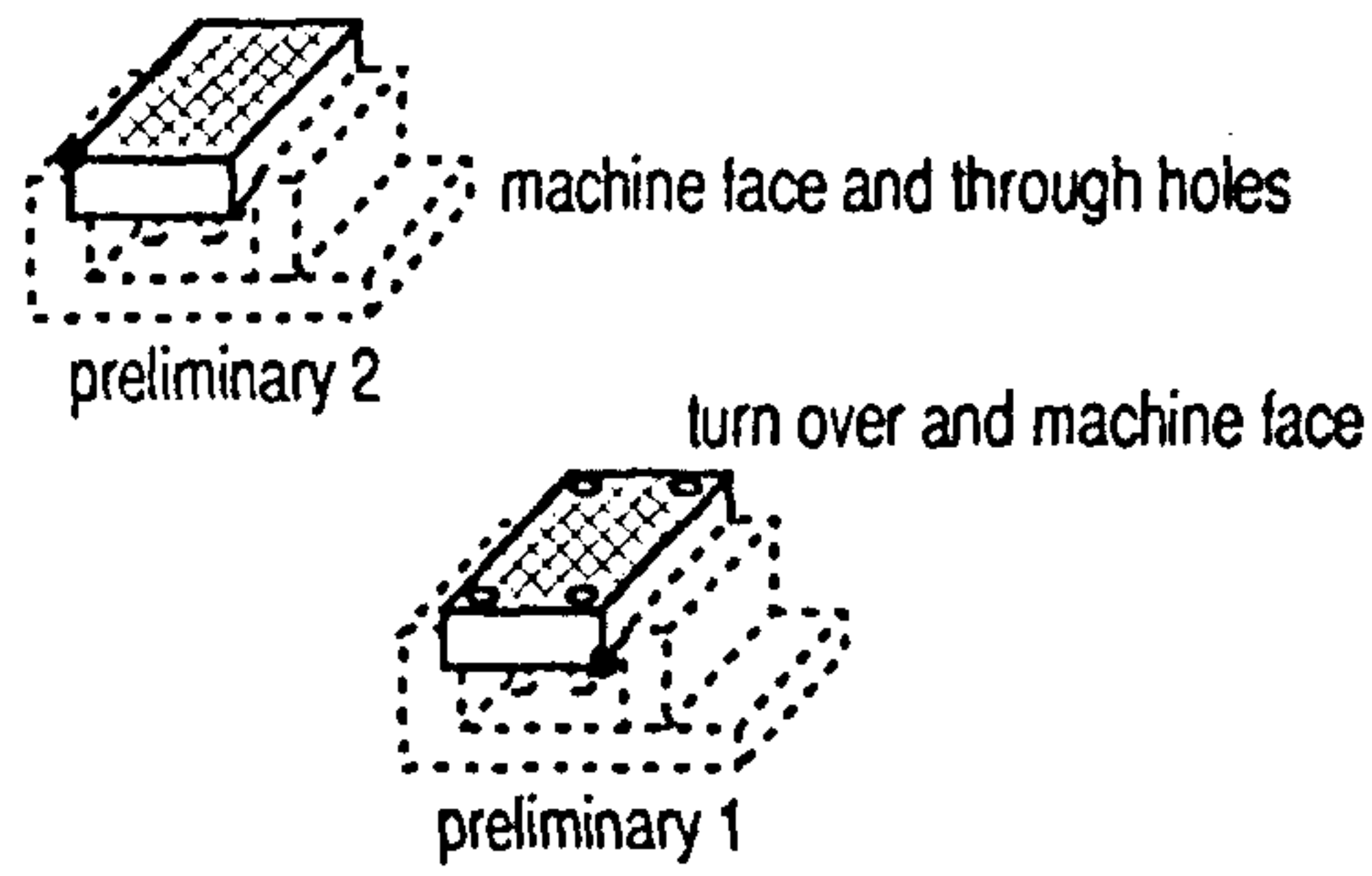
The fixturing strategies represented in the research have provided a basis from which it has been shown that feature interactions, influencing fixture strategy selection, can be identified. This has been achieved, based on the geometric queries described in chapter 7.4 and the experimental work described in chapter 11.3.4. Further work, addressing more complex strategies, will be required in future research. A paper, describing the research on fixturing strategies and geometric queries in setup planning, has been published in the International Journal of Production Research [156].

Machining rules provide control over the performance of machining activities. This is an area of research where the use of knowledge representation techniques are being explored extensively as described in the literature survey. Machining rules, in the author's research, provide a simple basis for control over the functions of the exploratory planner, such that the Close Coupling of Machine Planning functions to the PDM could be investigated. Knowledge representation has not been pursued as an issue. The ADA programming language has been an appropriate tool to represent simple control rules, and rules for setup planning, operation sequencing, tool selection and cutting parameter determination have been implemented using ADA function calls which are listed in appendix 5.9, and used as described in the following sections.

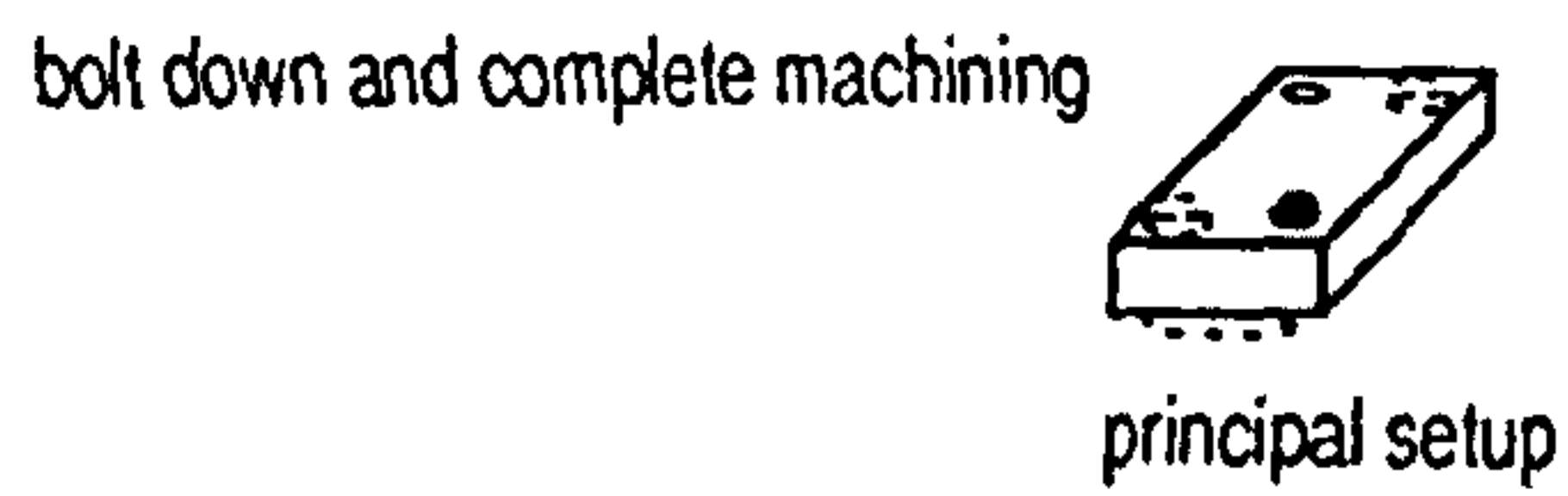




(a)  
Side  
Clamping  
Strategy



(b)  
Through  
Clamping  
Strategy



(c)  
Down  
Clamping  
Strategy

• - indicates fixture datum position  
 ☒ - faces to machine before principal setup

Figure 9.2

### 9.3 THE FUNCTIONS TO IDENTIFY COMPONENT SETUPS

The *Identify Component Setups* activity analyses a workpiece, to decide on the sequence of setups and the operations which should be used in the manufacture of a component. This is a key problem in Machine Planning, with many aspects to be considered as discussed in chapter 6.3.3. The approach taken here follows the method argued at the end of chapter 6.3.3.

*Identify Component Setups* comprises four activities, MP11, MP12, MP13 and MP14, in the IDEF0 model of Machine Planning. These are illustrated in figure 9.1(b). The implementation of MP11, MP12, and MP13, *Identify features, Sequence Setups and Evaluate Features*, and their sub-activities, are detailed below. MP14, *Finalise Setup Sequence*, is included in the model to represent the reassessment of a setup sequence, should the initial planning route prove unsuccessful. This was not implemented in the experimental software as it was not essential to the exploration of the Closely Coupled Application.

#### 9.3.1 Identify Features

*Identify Features*, activity MP11, provides a breakdown of the workpiece into features. This requirement in planning has led to wide ranging research into feature recognition, as described in the literature survey. Choi [30,31], has achieved the recognition of elementary shapes such as slots and pockets, while hierarchies of patterns have been recognised with the PART system [140], which makes some progress towards the recognition of interacting features. The end point of this activity is to provide the feature process data which is needed in planning. The author's research has explored the representation of such data in a PDM, through

the use of pre-defined features, as argued in chapter 6.2.4. The software implementation of the *Identify Features* activity has extracted feature data from a Product Model containing a component description built using pre-defined, parametric features, as described in chapter 10.4.4. The relationship of the implemented approach to feature recognition is proposed in the Machine Planning IDEF0 model in appendix 2.

In the experimental implementation, the machining feature process data, as defined in chapter 6.4, is extracted directly from the component definition in the Product Model. This has been done using the menu options explained in chapter 10.5.2, while a full description of the use of the software can be found in appendix 6. The feature process data, and the geometric transformations which have been applied to each feature, are extracted. The transforms are applied to the feature process data to identify the position of each feature with respect to the global coordinate system; and the orientation of the spindle axis directions, fixture face normal directions, and tool approach directions. The author's implementation has used the orthogonal axes directions of +x, -x, +y, -y, +z, -z as a sufficient basis for the experimentation, to demonstrate links from feature position and orientation data in the PDM to the Machine Planner. This meets the requirements of feature orientation, established in chapter 7.3.2, and provides a basis for the identification of feature positional data relationships, as argued in chapter 7.3.1.

The stock and component geometry data, captured in the component level planned process data structure defined in chapter 6.5, have also been extracted from the Product Model, as part of the *Identify Features* activity. This is needed when identifying relative feature positions, as discussed in chapter 7.3. Material specification data has also been extracted to be used in the calculation of processing data, described in section 9.3.3.2. These data elements have also been extracted through the implementation option '5' described in chapter 10.5.2, while the details of the



software link to the PDM are listed in appendix 5.1. The Product Data Model has been constructed to take full advantage of the use of parameter values, therefore limiting the need for geometric analysis where possible. The length, width and depth of the component have been extracted from the general parameter values and the stock overall dimensions have been found in the same way. It is recognised that more complex geometries would require greater levels of geometric analysis.

In addition to machining feature data, a Machine Planner requires data on fixturing features, as discussed in chapter 6.2. In the software implementation, possible fixture features have been restricted to be either through holes and Face features. These have been identified from the names of features contained in the Product Model component description, explained in chapter 10.4.4. Hence it has been a requirement that features which can be used for fixturing, must contain either "through hole", or "face" in their name. This has overcome the need to explore fixture feature recognition, which would have detracted from the main aim of the research. In the software implementation, the machining feature names are checked at the start of option 7, described chapter 10.5.2 and categorised as "through hole", "blind hole" "face" or "other" features. These are then used in fixture strategy selection described in section 9.3.2.4.

The identification of the position of features, with respect to the component coordinate axes, a requirement established in chapter 7.3.1, is also included in the experimental implementation. The volume of material to be removed from the workpiece is also calculated, a requirement described in chapter 6.3.3 and used in fixture strategy selection described in section 9.3.2.4. These software procedures are also called from option 7, described chapter 10.5.2, while the software link to the PDM which calls the implemented software is listed in appendix 5.1.

### 9.3.2 Sequence Setups

*Sequence Setups*, activity MP12, takes the workpiece description, including its breakdown into machining features, and decides on an appropriate sequence of setups and the features on the workpiece which should be used for fixture and datum setting. The author has chosen to implement a group of procedures which assess a component description against a number of fixturing strategies to produce a setup sequence.

Worldwide research into the automation of the *Sequence Setups* activity is still at an early stage. Armstrong [7] used material removal capability as a basis for sequencing, while Joshi [73] showed that spindle axis directions could be used for grouping. Darvishi [40], Iwata [69], and Roy [125] have explored knowledge representation issues, while Willis [149] and Kanumury [76] discuss the use of fixture strategies. Halvei [56] and Boerma [17] consider tolerance analysis of the component to decide on a sequence of setups.

In the author's work it was important to explore the PDM, as a source of product data which is appropriate to a Machine Planning Application. The *Sequence Setups* implementation provided the means to explore the use of the feature level process data, as defined in chapter 6.4; the use of the component level planned processed data, from the definition in chapter 6.5; and the use of feature positional and interaction data, using the methods chosen in chapter 7.3 and 7.4.

Five sub-activities, *Group Features by Spindle Axis Direction*, *Identify Geometric Relationships*, *Apply Setup Rules*, *Select Fixturing Strategy* and *Update Setup Data*, meet the necessary and sufficient scope of the *Sequence Setups* implementation, and are described below. These sub-activities form activities MP121 -> MP125 in the IDEF0 model and are illustrated in figure 9.1(c). Appendix 5.2 provides a list-

ing of the ADA code used in the "comp\_setups\_sequ\_method" procedure, within which the functions to perform the *Sequence Setups* activity are contained.

### 9.3.2.1 Group Features by Spindle Axis Direction

This implementation uses the Spindle Axis Directions of each feature to associate features with setups. Cases where a feature has multiple spindle axis directions have been catered for by creating duplicate features, one to represent each possible spindle axis direction, a requirement identified in chapter 7.3.2. When a duplicate feature is created, new reference axes for the feature have been defined as part of the implemented procedure. The procedure used in the implementation is contained in the ADA code listed in appendix 5.2.

### 9.3.2.2 Identify Geometric Relationships

It was initially assumed that the *Identify Geometric Relationships* would form the basic activity in which all the geometric relationships required in setup planning would be identified. However, geometric relationships, between features, are relative to some point in a coordinate system. They cannot, therefore, all be assessed at one stage in the overall planning procedure, unless each reference point and coordinate system is known. For example, the relative position of features to a datum cannot be identified, until the datum position is known relative to the setup axes. Therefore the implementation of this relationship cannot take place until after the fixturing strategy to be used is known.

The implementation of *Identify Geometric Relationships* identifies the position of features in a setup, relative to the setup coordinate axes, a requirement discussed in chapter 7.3.1. The remaining geometric relationships, discussed in chapter 7, are implemented within subsequent functions and are introduced progressively through



this chapter. Geometric relationship checks are contained in the ADA code listings in appendix 5.2, 5.3 and 5.4.

### 9.3.2.3 Apply Setup Rules

The *Apply Setup Rules* activity calls, and uses, the setup rules listed in appendix 5.9. These provide the technological constraint for fixture strategy selection. The application of the rules result in the principal setup, or the key setup to be fixtured, which is then passed to the *Select Fixturing Strategy* activity.

In the software implementation, this activity is represented by a number of functions and procedures, as can be seen in appendix 5.2. The rules called are limited to a necessary minimum, but provide a basis from which each setup can be checked to identify the principal setup. The procedures implemented assess the setups and reorder them, bringing the principal setup to the head of the list. This is firstly done by counting the number of features in each setup. The setup with the highest number is brought to the head of the list. Then the volume to be removed in each setup is calculated. If any setup has a high volume to be removed, based on a setup rule, then it is brought to the head of the list. The third check counts the number of features in each setup with tight tolerances, again based on a setup rule. The setup with the highest number is brought to the head of the list.

Having performed the actions described above, the setup at the head of the setup list becomes the principal setup. At this stage, each feature in each setup, is checked to identify its general suitability for fixturing. This has been done by firstly categorising features, dependent on their relative position in the setup, as being "central", "mid," or "outer". A central feature's position is at  $x=0$ ,  $y=0$  relative to the setup coordinate axes. An outer feature has been defined as being in the outer third of the setup surface. With this information, and the feature types al-

ready identified as described earlier, the setup rules which apply to fixturing are called. Face features are flagged as suitable for Primary Location; Face features and through holes are flagged as suitable for clamping; through and blind holes with a diameter greater than 6 mm, and categorised as "outer", are flagged as suitable for Secondary Location as are Face features.

The final action performed before fixture strategy Selection was then to identify the positions of each setup relative to the principal setup, as described in chapter 7.3.3.

The ability to identify an appropriate principal setup was limited by the lack of inter-feature dimensional data in the implementation, and the need for a close analysis of the rules which influence setup planning. Irrespective of these limitations, the implementation has provided a basis for the selection of a fixturing strategy, which has enabled the Close Coupling of a Machine Planner to be investigated.

#### 9.3.2.4 Select Fixturing Strategy

The *Select Fixturing Strategy* implementation has been used to explore the use of geometric queries, as discussed in chapter 7.4, to identify feature interaction data, and to explore the PDM as a mechanism for the provision of product data for planning, as discussed in chapters 5 and 6. The implementation is listed as part of the ADA code in appendix 5.2.

In the software implementation, Side Clamping is selected if light machining, only, is required. If this is not the case, then Through Clamping is tried. Its suitability is dictated, at this stage, by the availability of through holes, which have passed the setup rules. If Through Clamping is also found to be unsuitable, then Down Clamping is tried. Software representations of each of these three strategies,

described in section 9.2.3, have been implemented.

A listing of the ADA code used in the definition of the Through Clamping Strategy is provided in appendix 5. This shows that the principal setup is firstly checked for suitable holes, using the results of the *Apply Setup Rules* activity described in section 9.3.2.3. Each hole found is then checked using geometric queries as argued in chapter 7.4.2, to ensure that fixtures will not overlap any features to be machined.

The next stage of the Through Clamping Strategy implementation is then dependent on the number of suitable holes which have been found. A minimum of three holes are required in the implementation, two for clamping and one as a datum. Assuming this stage is passed successfully, then preliminary setups, datum and fixture features are found as part of the strategy, as described in section 9.2.3; feature positions are reset relative to the fixture datum, as described in chapter 7.3.1; and the side Face features are converted to profiles to suit the fixturing strategy, as discussed in chapter 7.3.2.

The ADA code, used to represent the geometric query which checks around a feature for interactions, is listed in appendix 5. Each of the implemented fixturing strategies, have been tested in the experiments described in chapter 11.3.4, which show that the approach taken is successful in providing a basis to check feature interactions. Setup plans generated through the use of each implemented fixturing strategy has been documented in appendix 7.4.

#### 9.3.2.5 Update Setup Data

The *Update Setup Data* implementation, forms the final part of the ADA procedure listed in appendix 5.2. The implemented procedures follow from the duplication of features, performed in the *Group Features by Spindle Axis Direction* activity



described above. Such features must to be removed, and this action is performed at this stage. The procedures check the name of each feature, in succession throughout a setup, against the feature names in subsequent setups. Where duplication is found the latter feature is removed. This procedure is repeated for each setup in turn. As a result of this deletion some setups may have no features to be machined. Where this has occurred the setup has also been deleted from the sequence.

There is still a need to pursue an implementation of updates to setup geometry, as argued in chapter 7.3.4. This has not been pursued in the time available for the work, but no fundamental problem in the implementation of this requirement can be envisaged.

### 9.3.3 Evaluate Features

Feature recognition research generally aims to identify geometry of significance for manufacture. The *Evaluate Features* activity, MP13, takes as input the setup sequence, and evaluates each machining feature, to identify the operations which are needed to machine it, and the possible tools which can be used in each operation's manufacture. The IDEF0 activity MP13, provides a view of how the pre-defined and recognised features approaches may be combined once setup level data has been identified. By using pre-defined features in the implementation (MP133), the operation data required to identify cutting tools can be extracted from each pre-defined feature and hence, operation data can be found directly from the Product Model rather than requiring a recognition process.

The implementation of *Evaluate Features*, activity MP13, then contains, within MP133, an implementation of *Extract Feature Operations* (MP1331), and *Identify Cutting Tools* (MP1332), and an implementation of *Calculate Processing Data* (MP134). These activities extract operation data from each pre-defined feature, and select possible tools and cutting parameters for each of the defined operations, as described in the following sub-sections. These functions are called from the link to the PDM described in chapter 10.5.2. The ADA code, which provides these function calls, is represented under variant 8 in appendix 5.1.

The implementation of these functions have provided a basis from which to prove the operation data structure in the PDM. Each experiment, described in chapter 11, has involved tool and processing data calculation, and the machining instructions and part programs, provided in appendix 7, document the data resulting from these.

### 9.3.3.1 Identify Cutting Tools

This activity take data from an operation and identifies the possible tools which can be used in its manufacture. Tool selection research for milling operations, by Melkote [96], has addressed the assessment of tool parameters for specific machining operations. Significantly, however, it has been recognised by Carlier [24] that specific tools should not be chosen for an operation without regard to the other operations to be performed during the machining sequence. As discussed in chapter 6.3.1, a tool selection procedure has been implemented to follow Carlier's approach. A listing of ADA code used in the procedure is provided in appendix 5.5.

Selecting tools to machine an operation involves the assessment of a wide range of cutters. In the experimental implementation, it has been assumed that only cutters with inserted tips would be used, except in the case of drills, where high speed steel has been assumed. The limitation to the use of inserted carbide provided a

restriction on the experimental capability when wishing to include the use of slot drills. To overcome this, solid carbide slot drills have also been used, but they have been assumed, in the experimental software, to have an inserted tip. This has enabled a wider exploration of the Close Coupling of a Machine Planner, without the need for major changes to the underlying software implementation.

The *Identify Cutting Tools* procedure has provided a basis for the assessment of the operation data structure, defined in the PDM, and has provided data which has been used in operation sequencing and part program generation. In combination with the range of implemented Machine Planning functions, this has provided a valuable aid to the assessment of the Closely Coupled Machine Planner.

### 9.3.3.2 Calculate Processing Data

The role of this activity is to generate speed and feed rate information, and to identify cut depths and cut widths, appropriate for use on the machine under consideration. Computer based cutting parameter determination generally uses either machinability data tables, such as those by Metcut [99] or an extended form of Taylor's tool life equation [100]. The latter method provides some flexibility by allowing the user some choice in selecting required tool life values, and provides the potential for more comprehensive methods of cutting data calculation in future systems. The need in the author's research was to calculate cutting parameters which could be used in part programs, and which in turn could be used to machine components, confirming the PDM as an appropriate mechanism for the Close Coupling of a Machine Planner, rather than research improved methods for cutting parameter determination.



A listing of the key elements of the ADA code used in the *Calculate Processing Data* implementation is provided in appendix 5.6. The implementation was based on the use of data from the Production Engineering Research Association (PERA) [100], who provided exponent values for a tool life equation for milling grade 17 grey cast iron. The use of the implemented software gave problems in the relatively high speed and feed rate data which were generated. However, components could still be produced on the Wadkin V4-6, using the data, which was considered to be sufficient for the purposes of this research. Other machining situations were covered by the use of tabular data from Metcut Research Associates [99].

As in the case of the *Identify Cutting Tools* implementation above, this procedure has provided a basis for the assessment of the operation data structure, described in chapter 6.4, and has contributed to the overall ability to explore the Closely Coupled Machine Planner.

#### 9.4 THE SEQUENCE OPERATIONS FUNCTION

*Sequence Operations*, MP3, takes as input the setup sequence, extracting a list of operations for each setup, where possible tools for each operation have been identified. The activity then produces a sequence of operations, identifying the specific tool which should be used in each case. The constraints on operation sequencing have been discussed in chapter 6.3.2. Joshi [73] has identified precedence as a constraint, while Carrier [24] has identified technological constraints and Chen [29], considering turning operations included cost considerations in his sequencing method. This research has implemented a procedure which enables the Close Coupling of the MP Application to be explored by generating operation plans for a setup, rather than attempting to advance the understanding of operation sequencing in its own right.

A listing of the ADA code for the operation sequencing procedure is provided in appendix 5. The implementation has been divided into two parts to meet the requirements discussed in chapter 6.3.2. These parts being the grouping of operations into sub-groups, followed by the selection of a specific sequence of operations with related tooling, minimising the time taken to machine the setup. The first part of the implementation follows from the operation sequencing rules listed in appendix 5.9. These are based firstly on performing roughing operations before finishing operations and secondly on performing operations in the order of facing, then end milling and slotting, then drilling.

The second part of the implementation takes into account the influence of cutter selection on sequencing operations. This has been done by sequencing each group of operations on the basis of decreasing diameter constraints. This produces a list which has a trend towards smaller cutter diameters moving towards the end of the list. Each possible cutter is then checked, working backwards through the list of operations, to identify the cutter which will take the least time to machine the operation. This takes into account the time required to change the tool in the spindle. This produces a sequence, based on minimising the time taken to machine the setup.

The effectiveness of the implementation could be improved by the inclusion of geometric precedence checks, as discussed in chapter 6.2.3. A comparison of inter-feature tolerances on an operation sequence is also needed as discussed in chapter 8. Nevertheless, the implemented procedure has produced operation plans, which have been used successfully in the production of part programs, as discussed in chapter 11 and represented in appendix 7.

## 9.5 THE FUNCTIONS USED TO GENERATE PART PROGRAMS

The Part Programs generated in this research have been produced for a GE2000 controller [50] on a Wadkin V4/6 machining centre. The extraction of a part program must ensure that the correct parameters are sent to the tool path description of each operation in turn, such that a segment of a part program is generated for each operation. It is necessary to produce NC code to start the program, set the fixture datum to zero, using a touch trigger probe, and end the program once code has been generated for each of the operations.

Input from the user is requested to identify the approximate position of each fixture datum, relative to the machine table, as discussed in chapter 7.3.1. This data is then used, when generating the part programs, to control probing routines which locate the actual position of the fixture datum for each setup, and set fixture offset values in the machine controller accordingly.

The utilities used to produce individual pieces of NC code, for operations, have been generated through separate work in the research laboratory by M.J. Corrigan [36]. The Author's research has used these routines to generate part programs from an operation sequence. These part programs have then been used to produce components which have been checked with the Product Model representation and used in the downstream Inspection Planning and Manufacturing Data Analysis Applications, described in chapter 5.5.

A listing of the ADA code used in the extraction of machine control code is provided in appendix 5. This has been used to generate part programs, as described in the experimental work of chapter 11. These part programs have then been successfully used to machine the testpiece components on the Wadkin V4-6 machining centre.



This chapter has explained and defined the Machine Planner functions, strategies, and rules which have been implemented in order to carry out the research exploration. The following chapter explains how a Product Model can be built from the PDM, and how the implemented Machine Planning functions can be linked to the PDM to produce the Closely Coupled Application which has then been explored experimentally as described in chapter 11.

## CHAPTER 10

### THE REALISATION OF AN EXPERIMENTAL MACHINE PLANNING APPLICATION IN THE PRODUCT MODEL ENVIRONMENT.

#### 10.1 INTRODUCTION.

This chapter describes the work performed to realise a Closely Coupled Machine Planning Application in the Product Modelling environment. It explains the linkage of the Machine Planning software, described in chapter 9, to the Product Data Model described in chapters 5 and 6. Appendix 4 documents data models which have been populated, and used in the experimental exploration of the Closely Coupled Application, while appendix 6 provides a guide to the use of the experimental software. The experiments performed, using the benchmark component, to explore the success of the approach, are described in chapter 11.

The aspects of the research described in this chapter cover the experimental environment, the population of Product Models, and the linking of Applications to the Product Data Model. The exploration of the Closely Coupled Machine Planning Application has been performed in parallel with the Inspection Planning, Manufacturing Data Analysis, Cutter Path Generation, Relationship Graph and Geometry research of the ISS project, described in chapter 4. The realisation of an exploratory integrated system has been influenced by the rate of development of ideas across the research team and by the scale of work involved in drawing together the results of the integration issues being explored by the researchers. This has resulted in an exploration of integration where the integrating framework, the PDM, as well as the Applications, has been evolving. Some aspects of the integration work have

therefore been more highly developed than others, dependent on the progress achieved at particular stages in the research programme.

## 10.2 THE EXPERIMENTAL ENVIRONMENT

The work to realise an integrated experimental Machine Planning Application has been performed in an environment consisting of three main parts; computing, machining and a project environment in which a number of co-researchers interacted.

The computing environment consisted of a software programming environment operating on a network of SUN workstations. The network being based on a SUN 3/160 fileserver linked to SUN 3/50 diskless nodes. The software programming environment consisted of two parts; an experimental product modelling tool and a software programming language. The experimental product modelling tool used was the Structure Editor (SE) [16], described in chapter 4 and appendix 1. This provided the means for the integration exploration, by providing the facilities to build Product Data Models, instantiate particular Product Models and to generate interface routines to Applications. The software programming language used was ADA [66]. This was chosen by the ISS project as an appropriate tool for data modelling because of its strong typing and suitability for the integration of large scale software projects.

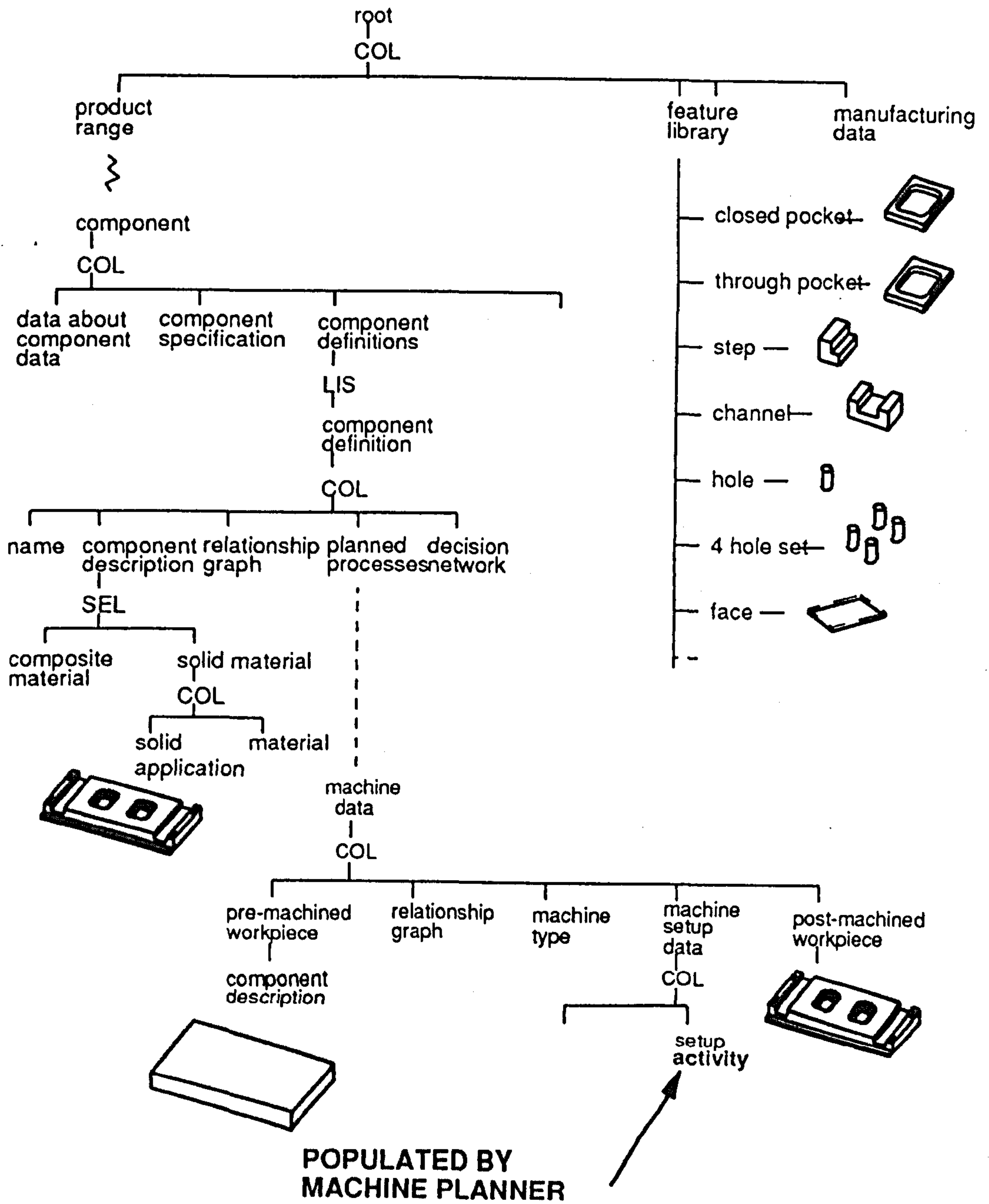
The machining environment consisted of a Wadkin V4/6 machining centre and a Ferranti coordinate measuring machine. The Machine Planning research has been targeted at the Wadkin V4/6 and the part programs, generated in the experiments described in chapter 11, have been tested using this machine. A range of cutters and fixtures were also available for use. The cutter range was a 63 mm diameter



face mill; 50, 40, 25, and 12 mm diameter end mills; 20, 16, 12, 8 and 4 mm slot drills and 12, 10 and 6 mm high speed steel twist drills. Except in the case of the twist drills, carbide tooling was used. A vice, bolts and clamps were available for fixturing, which were appropriate for use in the exploration of fixturing strategy selection.

Links with other workers on the ISS project was the third main part of the experimental environment. The principal co-workers involved in the programme, whose work impinged on the Author's contribution were P.Bell (Cutter Path Generation), M.J. Corrigan (Inspection Planning), L.Lee (Manufacturing Data Analysis), L.P.Wickens (Dimensions and Tolerances), D. Dunnington (Geometry) and A. McKay (Structure Editor).

In exploring integration in design and manufacture it was essential to ensure that the aspects being pursued by the various project members were co-ordinated and that each individual's research was contributing to a common integration goal. To achieve this, the researchers met regularly, to discuss the technical issues involved in integration, using the benchmark workpiece as a common focus for the problems being encountered, and as a basis for the exploration of their solution. The provision of formal relationships between the manufacturing functions being pursued were generated by the use of the IDEF0 modelling methodology.



A Product Model Framework

Figure 10.1

### **10.3 THE POPULATION OF A PRODUCT MODEL.**

#### **10.3.1 A Framework for a Product Model**

The SE has been used as the tool for the construction of Product Data Models and the population of related Product Models to provide a base for the research exploration. To build a Product Model there is a range of data elements which need to be represented, but which are not specific to a product. In particular, features and manufacturing data fall into this category. Figure 10.1 illustrates the framework used in the construction of Product Models, providing manufacturing data on machine tools, and cutters, and a feature library.

#### **10.3.2 A Library of Machining Features**

The SE provides a parametric definition capability which has been used in the pre-definition of machining features, capturing the feature process data structure argued in chapter 6.4. Machining feature definitions have been built, within the feature library, and then used to build component descriptions. These component descriptions have, in turn, been used in the experiments discussed in chapter 11. Section 10.4 describes how these features have been defined and used in producing component descriptions, while machining features which have been defined in the feature library are represented in appendix 4.



### 10.3.3 The Solid Application

The Solid Application within the PDM provides the ability to define a component as a solid plus features, as illustrated in figure 10.1. This provides a representation mechanism with which a component solid can be defined, either without the use of features; partially defined with features; or be fully defined using features. The experimental realisation of the author's research has used the last of the above options. However, the alternative capabilities of the mechanism open up future possibilities for the exploration of combined feature recognition and pre-definition techniques. Component descriptions, using this facility, have been defined as described in section 10.4.4 of this chapter, and have provided a basis for the experimental work described in chapter 11.

### 10.3.4 Manufacturing Data.

It is essential to be able to access manufacturing data while working within an integrated environment. This data is needed for planning decisions, but is not specific to a particular product. Hence, manufacturing data has been represented as separate to the Product Model, but as a part of the Product Model framework. By taking this approach it has been possible to link from a Product Model directly to the relevant manufacturing data for a product.

The manufacturing data represented in the research includes machine data and cutting tool data. The description of machine and cutter data, as represented in the Product Model framework, is documented in appendix 4, and has provided data to the procedures described in chapter 9.3.3, which identifying cutting tools and calculating processing data.

## 10.4 THE REPRESENTATION OF MACHINING FEATURES WITHIN THE PRODUCT MODEL ENVIRONMENT.

### 10.4.1 The Pre-definition of Machining Features.

The Structure Editor provides a capability for the representation of parametric definitions, as described in appendix 1. This capability has been used in the research to build a range of machining features which were then used to explore the feature level *process data* sub-set of the Product Data Model, argued in chapter 6.4.

A pre-defined feature consists of a list of parameters and a populated data structure, based on the feature description, argued in general in chapter 5.4.1 and containing the detailed feature level *process data* sub-set, argued in chapter 6.4. Any parameters which are used in the population of the feature are included in the parameter list. These parameters can then be given specific values when the feature is used to build a component description. Parameters can be defined as Numeric Expressions as described in appendix 1, and these can be used to capture dimensional tolerances, as discussed in chapter 8.3.1.

Surface parameters, such as roughness, can be defined as parameter values in their own right. This provides a basis from which Machine Planning assessments can be made. The definition of surface tolerances in this way does not identify to which surface the tolerance is linked, and only the general tolerance requirement for the feature can be defined. A further problem with using feature parameters is that they cannot provide inter-feature tolerance data. The relationship graph work of Wickens [148] provides the potential to overcome some of these problems, although his work is not feature related. Chapter 8 has argued a potential approach to assessing

Relationship Graph data, in conjunction with feature data, to provide inter-feature tolerance data for Machine Planning.

#### **10.4.2 The Range of Machining Features Defined in the Library.**

A library of machining features which have been used in the building of component models was produced as illustrated in figure 10.1. The features described were the Through Pocket, the Closed Pocket, the Channel, the Step, the Through Hole, the Four Corner Holes, and the Face. The full definition of these, as represented in the SE, is provided in appendix 4. These machining features are of "negative" and "surface" categories, discussed in chapter 6.2.2. No "positive" features have been implemented as they seemed to be little advantage to be gained from this, which would aid the exploration of the feature level *process data* sub-set of the Product Data Model. Also the use of "positive" features were likely to introduce feature interactions which would require a level of analysis which is beyond the scope of this thesis. The use of these features in the experimental work is described in chapter 11.3.1 while the machining instructions generated from individual features are documented in appendix 2.

#### **10.4.3 General Parameter Representations**

Pre-defined features are parametrised functions, which can return particular data values in their data structure when they are evaluated. In addition to feature representation, the SE allows simple functions to be represented which can then be used in feature and component definitions. In this way, key values in component descriptions have been defined as parametrised functions. These include values for the component length, width and depth and the stock length width and depth. This has enabled these values to be readily used elsewhere in the component definition,



without the need to know their actual values. General parameters which have been represented and used in the research are provided in appendix 4.4.9.

The ability to represent general parameters has been used to represent finishing allowances and clean up allowances, such that their effect on operation dimensions were captured. Each implemented feature consists of roughing and finishing operations, with the two being related by the finishing allowance which is to be used. This finishing allowance has been represented as a general parameter which has then been available for inclusion in feature definitions.

In a similar way, a function to define a clean up allowance has been used to identify the minimum amount of material which should be removed from a stock surface to ensure a cleanly machined face. This has been used in the parameters of Face features, when they have been added to the component description, as described in the following section. The use of general parameters in component descriptions, can be seen in the following section, while their use in feature representations can be seen in appendix 4.4.

#### **10.4.4 Adding Features to a Component Description.**

The features described above have been used to produce component descriptions, which have then been used to explore the success of the approach taken in the research. Representative component descriptions are documented in appendix 4.5 and have been used and the experiments described in chapter 11.

Features applied to the component description have been given descriptive names, for identification, and are moved to their required position on the component model using geometric transformations. The parameters of each feature are given values and the description of the component is then evaluated so that the actual values of

each parameter are passed throughout the structure. An illustration of a component description of a pocket applied to a block is shown below.

```
test_piece : Geometry:Transformed solid:
  block with origin at centre
    x = body length
    y = body width
    z = body thickness
  AT
    mx = body length / 2.0,
    my = body width / 2.0,
    mz =- ( body thickness / 2.0 )
  +
left top pocket : Pocket with axes and top surf at (0,0,0)
  x_dim= 50.0
  y_dim= 40.0
  z_dim= 5.0
  corner rad= 7.0
  fillet rad= 1.0
  surf fin (Cla microns): 2.5
  AT
    mx = 50.0,
    my = 37.5
```

where general parameter values:

```
body length = 150
body width  = 75
body thickness = 25
(dimensions are in mm)
```

Face features have been used in the research as a means of providing the link from the component description to the stock description. Hence, each face of the rectangular component, is defined as having a Face feature associated with it. The difference between the stock and component dimensions are therefore captured in the Face feature description, which then provides the operation data required to identify how each face is to be machined. This approach has been used to build a complete representation of a component with its six faces providing the link to the stock description as shown below. This forms the basic workpiece representation to which negative features can be added, and has provided a route to the exploration of the use of fixturing strategies and geometric queries in the setup planning part of

the Machine Planner implementation as described in chapter 9.2.3. In the description of the Glacier workpiece, represented in appendix 4.5.2, the block with its six faces has been represented as a "machined block" in the feature library, which reduced the complexity of the component description.

test\_block : Geometry:Transformed solid:

block with origin at centre

x = body length

y = body width

z = body thickness

AT

mx = body length / 2.0,

my = body width / 2.0,

mz =- ( body thickness / 2.0 )

+

top face : face block with origin at centre of bottom face

x-dim= stock\_length

y-dim= stock\_width

z-dim= rem\_stock\_height

surf fin (Cla microns): 2.5

AT

mx =( stock\_length / 2.0 - rem\_stock\_length),

my =( stock\_width / 2.0 - min\_stock)

bottom face : face block with origin at centre of bottom face

x-dim= stock\_length

y-dim= stock\_width

z-dim= min\_stock

surf fin (Cla microns): 2.5

AT

rx = 180.0,

mx = ( stock\_length / 2.0 - rem\_stock\_length),

my = ( stock\_width / 2.0 - min\_stock),

mz = - ( body thickness )

back face : face block with origin at centre of bottom face

x-dim= stock\_length

y-dim= stock height

z-dim= rem\_stock\_width

surf fin (Cla microns): 2.5

AT

rx =- ( 90.0 ),

mx =( stock\_length / 2.0 - rem\_stock\_length),

mz =- (( stock height / 2.0 - rem\_stock\_height) ),

my = body width

front face : face block with origin at centre of bottom face



```

x-dim= stock_length
y-dim= stock height
z-dim= min_stock
surf fin (Cla microns): 2.5

```

AT

```

rx = 90.0,
mx =( stock_length / 2.0 - rem_stock_length),
mz =- (( stock height / 2.0 - rem_stock_height) )

```

left face : face block with origin at centre of bottom face

```

x-dim= stock height
y-dim= stock_width
z-dim= rem_stock_length
surf fin (Cla microns): 2.5

```

AT

```

ry =- ( 90.0 ),
my =( stock_width / 2.0 - min_stock),
mz =- (( stock height / 2.0 - rem_stock_height) )

```

right face : face block with origin at centre of bottom face

```

x-dim= stock height
y-dim= stock_width
z-dim= min_stock
surf fin (Cla microns): 2.5

```

AT

```

ry = 90.0,
my =( stock_width / 2.0 - min_stock),
mz =- (( stock height / 2.0 - rem_stock_height) ),
mx = body length

```

where general parameters:

body length, body width, and body thickness are component dimensional parameters.

stock\_length, stock\_width, and stock height are stock dimensional parameters.

min\_stock = clean up allowance + finishing allowance.

rem\_stock\_length = stock\_length - (body length + min\_stock).

rem\_stock\_width = stock\_width - (body width + min\_stock).

rem\_stock\_height = stock height - (body thickness + min\_stock).

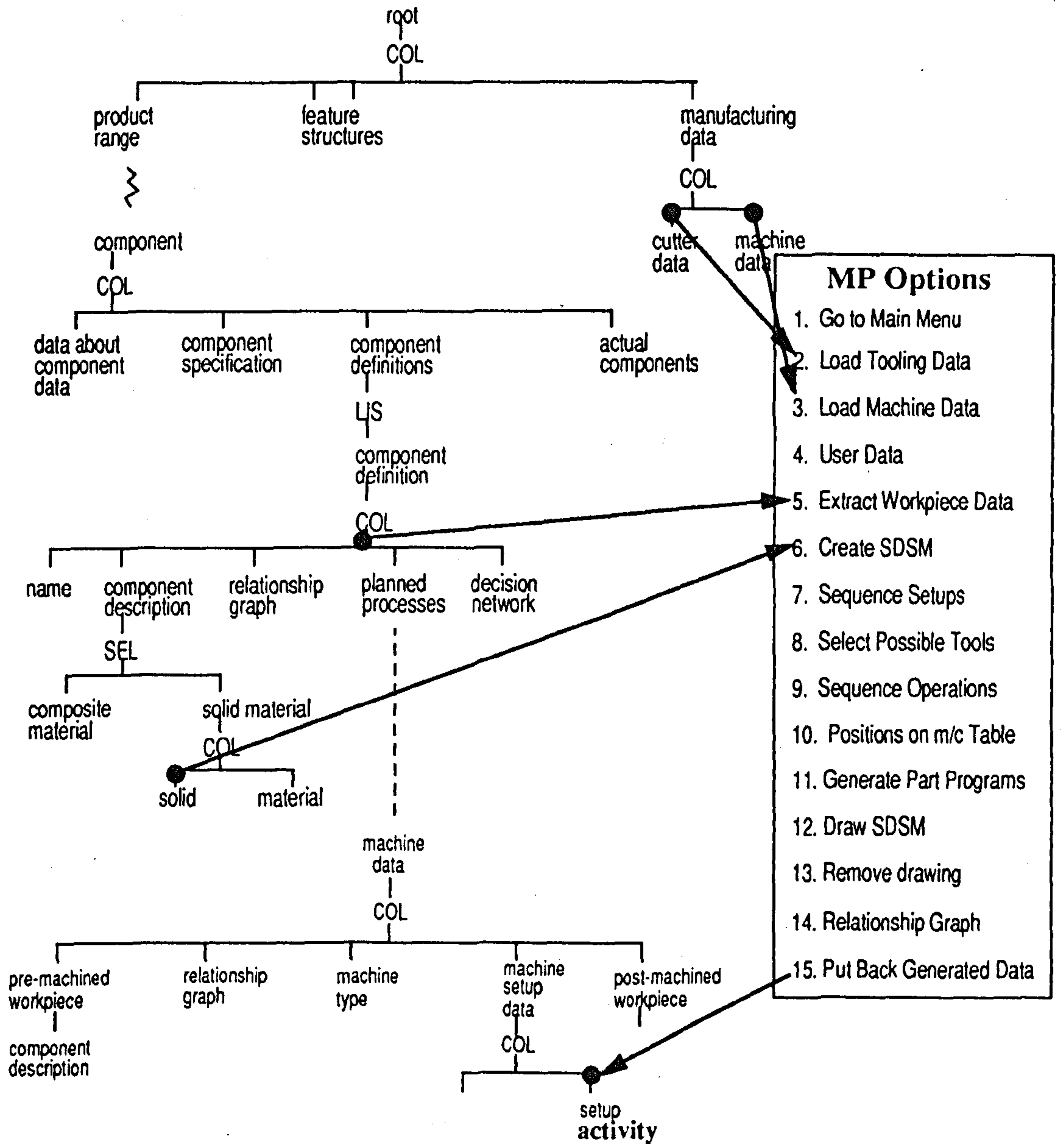
Component descriptions, such as the ones above, with their underlying data structures, provided the key source of product data into the Machine Planning Application. The links to the Application are described in the following section, while component descriptions for testpiece component are provided in appendix 4.5. Sample machining instructions and part programs, for testpieces are documented in appendix 7.3 and 7.4.

## **10.5 LINKING THE APPLICATION TO THE PRODUCT MODEL ENVIRONMENT.**

### **10.5.1 Interfacing to a Product Data Model.**

In order that the data in a Product Model can be used by a software Application, interfaces must be written which transfer the data in the Product Model into the form required by the Application. This must be done for each section of the PDM being used. Similarly, interfaces must be written to put data back into the Product Model.

The SE provides a facility which generates the skeleton of an interface. This skeleton provides ADA code which will traverse the data structure and extract values for the data contained within it. These interfaces must be edited to extract the appropriate data for the Application. Similarly, data can be transferred back to the Product Model. Such interfaces have been produced to link the experimental Machine Planning functions to the PDM, enabling the Closely Coupled Machine Planner to be used as described in the following section. The ADA code listing, showing the link from the PDM to the Machine Planning Application is documented in appendix 5.1.



**The Machine Planning Options  
in the Product Model Environment**

**Figure 10.2**



### 10.5.2 The Machine Planning Options in the Product Model Environment.

The Options listed below were linked to the Product Data Model to explore the integration of Machine Planning. The ADA code used is listed in appendix 5.1 and a software users guide is supplied in appendix 6. Figure 10.2 provides an illustration of the options, showing the Machine Planner linked to the relevant nodes in the PDM. From these nodes the product data structure is traversed to extract the data as required by the specific option. The italicised text in the list highlights the Machine Planning activities which are involved in particular options.

1. Go to Main Menu
2. Load tooling database
3. Load machine tool data
4. User data on tool life options
5. Extract workpiece data (*Identify Features*)
6. Create SDSM
7. Plan setups for the workpiece (*Identify Features and Sequence Setups*)
8. Identify possible tools for each operation (*Evaluate Features*)
9. Sequence the operations in each setup (*Sequence Operations*)
10. Relate fixture datums to the machine table (*Generate Machine Controller Code*)
11. Generate part programs (*Generate Machine Controller Code*)
12. Draw the SDSM model
13. Remove drawing
14. Relationship Graph
15. Put back planned process data

Options 2 -> 6 above required interfaces to be written to extract the data, needed for Machine Planning as discussed in chapters 5, 6, and 7, from the Product Model, and to link to the SDSM. Options 7 -> 11 provide the link to the Machine Planning functions described in chapter 9 while options 12 and 13 provide control over the visualisation of the SDSM model being used. Option 14 was introduced to explore links to the relationship Graph as described in chapter 8, while Option 15 allows data, which has been generated by the Closely Coupled Machine Planner to be put back into the Product Model.

### **10.5.3 The Effect of Changing a Product Data Model.**

Changes to the data structures in the PDM produced significant timescale problems in regenerating new working interfaces. Interfacing to the Product Data Model is a time consuming and complex task. It was not uncommon in the ISS project for researchers to spend three to six months generating interfaces to link their Applications. This was partly because of the number and complexity of the interfaces and, partly, because of the size of the software being produced. The executable file for the Machine Planning software, linked with the PDM was 2.7 Megabytes. The number and complexity of ADA packages involved in the software meant that early versions of ADA compilers gave problems. One compiler became unusable, while another although operating, could not be used in its debug mode. Debugging therefore became a very time consuming process. The most recent versions of ADA compilers appear to cope with our requirements, but the research software has apparently been operating at the bounds of compiler technology. The executable file for the full ISS software, linked to the PDM, was over 9 Megabytes.

Small changes to the structure of the PDM, which were restricted to the lower levels of a data structure, were less of a problem, as only the interfaces relating to that lower level needed to be changed, and instances of the PDM could be readily updated. However, when larger changes were made, not only were new interfaces required, but the data for specific instances of the Product Model had to be regenerated.

Changes to the PDM were not, therefore, made on a regular basis. However, with the range of research topics being pursued on the ISS project, the PDM could not remain static. Therefore, upgrades had to be made on several occasions. The evolution of the PDM, through these upgrades, is explained in the following section.

## **10.6 THE EVOLUTION OF THE PRODUCT DATA MODEL**

The definition of the Product Data Model, described in chapters 5 and 6, has evolved over a period of time. This evolution has been influenced by two earlier versions of the PDM, which have been constructed to test the progress of the research work. The research of the author has explored three major versions of a PDM, "feat.lam", "machplan.lam", and "pms3\_8.lam" to establish the product data structures described in chapters 5 and 6. The earliest version, "feat.lam", explored data structures to link a tooling data base to a tool selection procedure to cutter path generation. This involved the author in the exploration of operation data representations and links to machine tool and cutter data in order to select cutters and cutting parameters. The experimental work involved in this exploration is described in chapter 11.2.



The second version of the PDM, "machplan.lam", was used exclusively by the author, for the exploration of the integration of Machine Planning functions, without linking specifically to the work of the other members of the ISS team. The knowledge that further integration linkages could be made at a later stage was maintained through discussions with other team members. "Machplan.lam" was used to link operation data to features and to extend the operation data representation to include tool path data and operation constraint data. The feature process data representation was also added as a requirement for setup planning.

As well as the exploration of feature data "machplan.lam" was used to explore the addition of features to a component model and the subsequent extraction of the relevant data; to explore the introduction of component level planning data as a necessary provision of workpiece stock and component parameters for Machine Planning; and to explore the use of the SDSM as a medium to provide feature interaction data. "Machplan.lam" was used in experiments to generate machining instructions for individual features, features in a setup, components and in fixturing strategy selection experiments which are described in chapter 11.3. Features and component representations, built using this version, are provided in appendix 4.

The final version of the PDM, "pms3\_8.lam", has been argued in detail in chapters 5 and 6. This version has evolved from the evaluation of the "machplan.lam", capturing data requirements identified from it, as well as the requirements for integration which had been identified through interaction with the other members of the project team. The software representation of the final version of the PDM was constructed by A. McKay, and used by the author and the other members of the ISS research team. This enabled the author to demonstrate the Closely Coupled Machine Planner. It also enabled the author, in conjunction with the other members of the ISS project team, to show the level to which integration had been achieved across the project. The Glacier workpiece, experiment, described in chapter 11.4,

was used for this purpose and its component description, captured in the PDM is documented in appendix 4.5.2.

This chapter has shown how the Machine Planning functions, explained in chapter 9, have been linked to the Product Model environment to produce a Closely Coupled Application. This has provided a software base from which experiments have been performed to explore the success of the approach taken. These experiments are described in the chapter 11 and the resulting machining instructions and part programs are documented in appendix 7.

## CHAPTER 11

### A DISCUSSION OF THE EXPERIMENTS PERFORMED IN THE EXPLORATION OF THE CLOSELY COUPLED MACHINE PLANNER

#### 11.1 INTRODUCTION.

This chapter reports the author's experimental work which was performed as part of a sequence of demonstrations which allowed the researchers on the ISS programme to test the progress of their work. The experiments principally used variants of the Bolster plate workpiece, described in chapter 6.2, which was defined by the ISS research team as an appropriate means of drawing together the various research strands of the project. It also provided a basis for the discussion of research deliverables with the project's industrial collaborators and ACME review panel members.

The experimental work starts with an early experiment in integration, and then moves through a number of experiments where the data requirements for the Machine Planning Application were explored, in isolation from other manufacturing Applications. The final experiment captured the understanding gained from the earlier work, plus the input from the other members of the ISS research team to produce a demonstration of the level of integration which could be achieved between the MCG activities of the ISS research project.

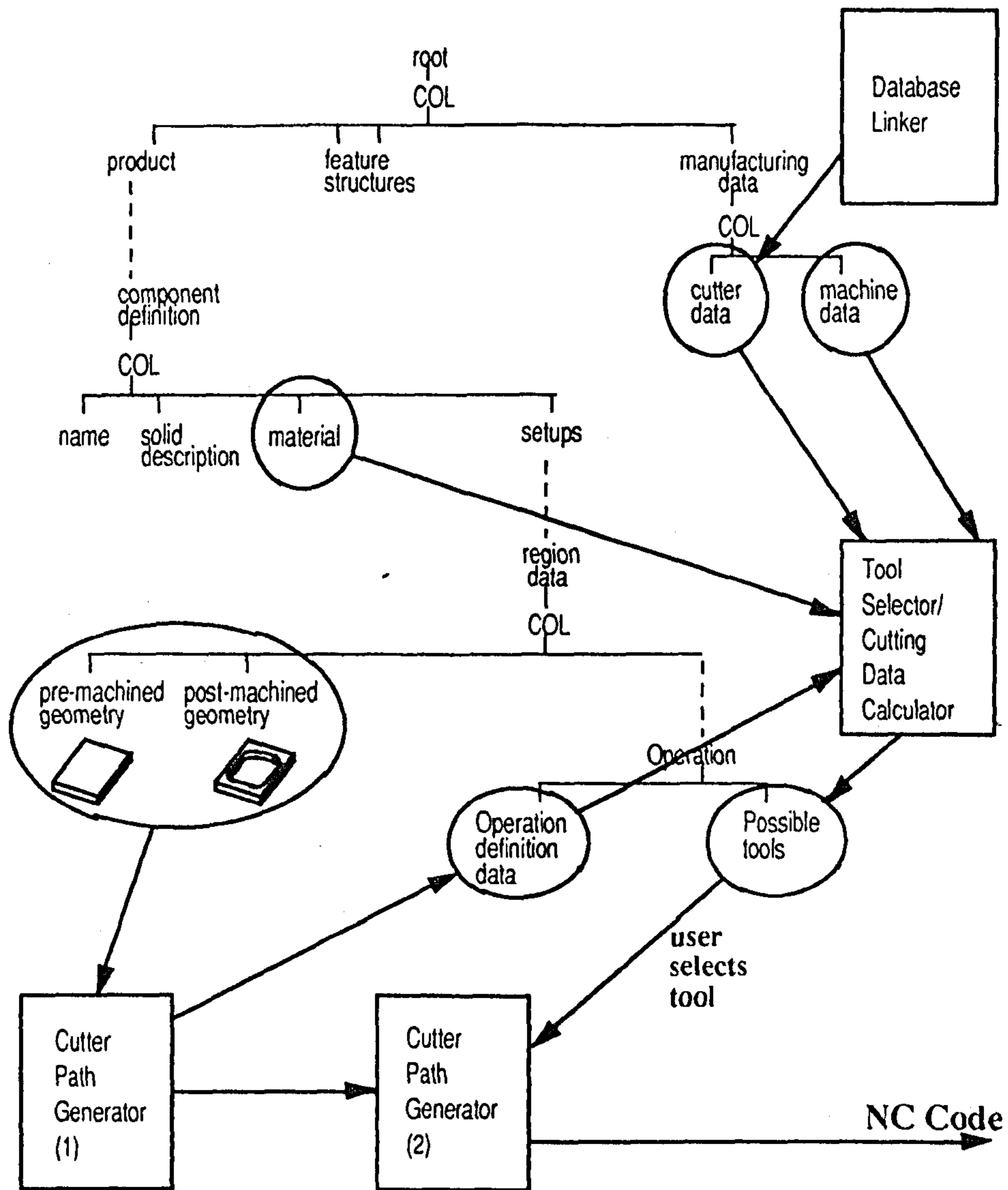


## 11.2 THE INTEGRATION OF A DATABASE, A TOOL SELECTION PROCEDURE AND A CUTTER PATH GENERATOR THROUGH A PDM

The experiment described in this section used the "feat.lam" PDM described in chapter 10.6, which contained a geometry structure, an Operation data structure, and the machine and tooling database structure. The Operation data structure contained the "Operation definition data", "constraints", "geometric data" and "possible tools" described in chapter 6.4. The experiment demonstrated that different Applications could be linked through the use of a PDM

Three separate software Applications, as illustrated in figure 11.1, were linked. The first Application was a database linker, produced by A. McKay [93], to extract tooling data from a proprietary database and to populate the tooling database of the Product Model framework. The second application was a cutter path generator, in two parts, produced by P. Bell [10]. Part one evaluated the geometric description, of a single machining region, to identify the Operation data required by the tool selection procedure. These were then used to populate the Operation data structure defined by the author. Part two, of the cutter path generator, having received back the tooling to be used via the Product Model, generated a cutter path to rough machine the Operation.

The third Application, produced by the author, was a tool selection procedure, as described in chapter 9.3.3.1, and incorporating the calculation of cutting parameters as described in chapter 9.3.3.2, with the exception of the tool path linkage. This procedure used the Operation data constraints generated by the cutter path generator, and manufacturing data as described in chapter 10.3.4, with the tooling database having been populated by the database linker. With this data the tool selection procedure identified possible tools for the Operation and the cutting parameters to be used. This data was returned to the tooling data fields of the Operation data

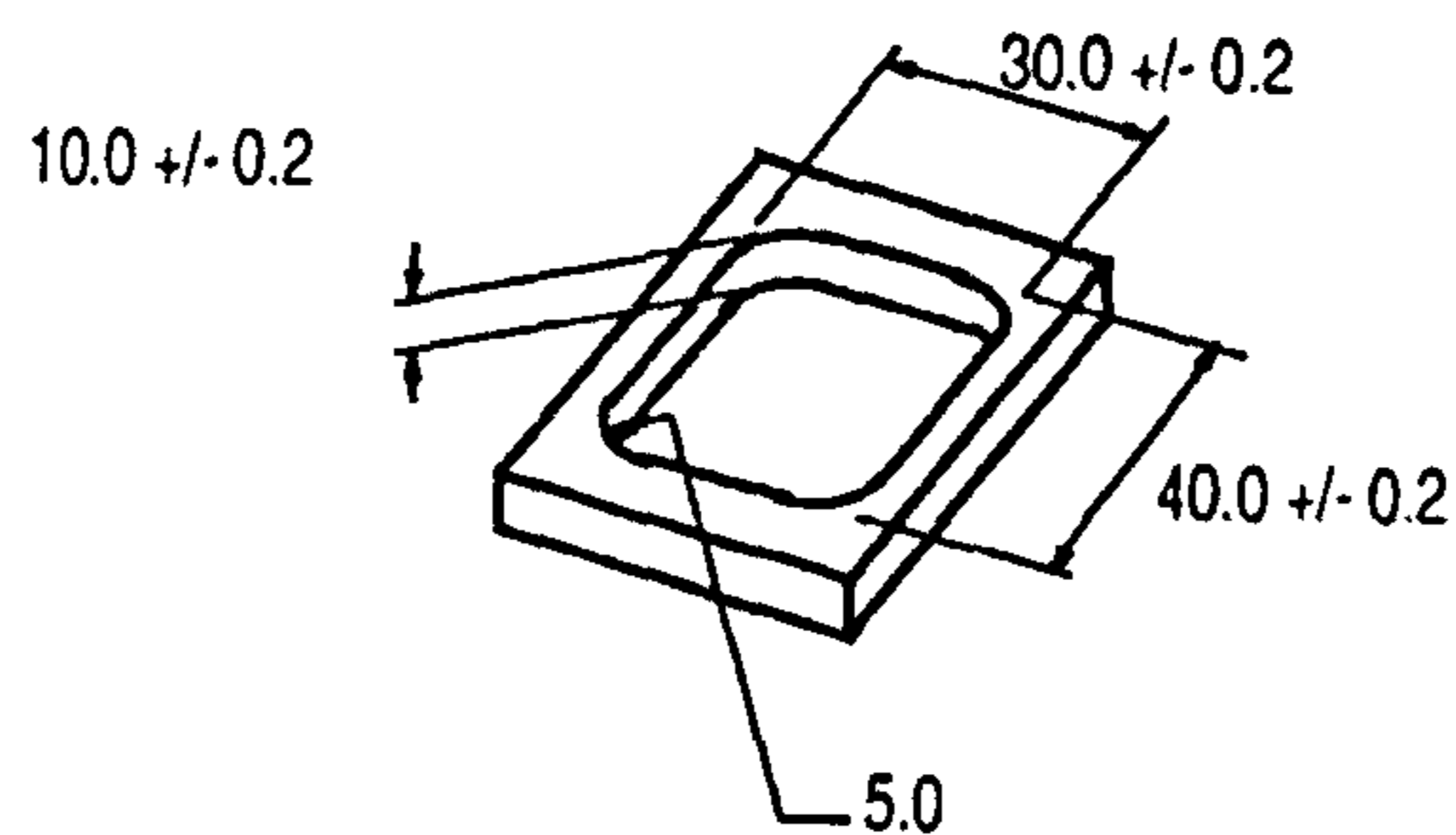


**The Integration of a Database, a Tool Selection Procedure and a Cutter Path Generator through a PDM**

**Figure 11.1**

sub-set of the Product Model, and then used by the second part of the cutter path generator.

This experiment demonstrated the first achievement of integration through the use of a PDM. It showed, using the component illustrated in figure 11.2, which was modelled in the PDM, that a set of possible tools and associated cutting data could be generated from the data in the Product Model and passed back into the data structure for subsequent use [155]. It was also shown, through the parallel work of



**Dimensions of the Test Pocket**

**Figure 11.2**

P. Bell [10], that the necessary Operation data for tool selection could be generated through geometric analysis and that the subsequently generated tooling data could be used for cutter path generation. Data structures, representing a tooling database, which could provide cutter data for tool selection were constructed by A. McKay and it was shown by her that these could be populated from a proprietary database [93].

It was also found that there was a significant time overhead in constructing, populating and interfacing to even a simple PDM. It took approximately six months to interface the exploratory tool selection software to the PDM and to populate a test Product Model to check and debug the interface. This was partly due to the learning needed for the techniques involved, as this was the first experiment of its type. However, in later experiments the improved understanding of the techniques required was counterbalanced by the increased complexity of the data structures and the increased time involved in populating an experimental Product Model framework. The time taken for each substantial change was in the order of three to six months. This was influenced by four factors: the scale of the change to the PDM and the resultant interfacing complexity; the clarity of understanding between the researchers linking to a new PDM and the need for revisions brought about by misunderstandings; the need to repopulate the Product Model framework with data;



and lastly the scale of the integrated software made debugging difficult.

Having proved the concept that separate software Applications could be linked through the use of the PDM, the next stage of the experimental work was to explore, in greater depth, the product data requirements for planning. The issues involved in extending the cutter path generation work were carried forward by P. Bell who pursued methods of analysing a component setup to attempt to break it down into machining regions and then to identify the Operation constraints. At this point the author chose to pursue the use of pre-defined features with associated tool path plans as this allowed a more direct, and certain, route to the exploration of a Closely Coupled Machine Planner

### **11.3 THE INTEGRATION OF THE MACHINE PLANNING APPLICATION THROUGH A PDM**

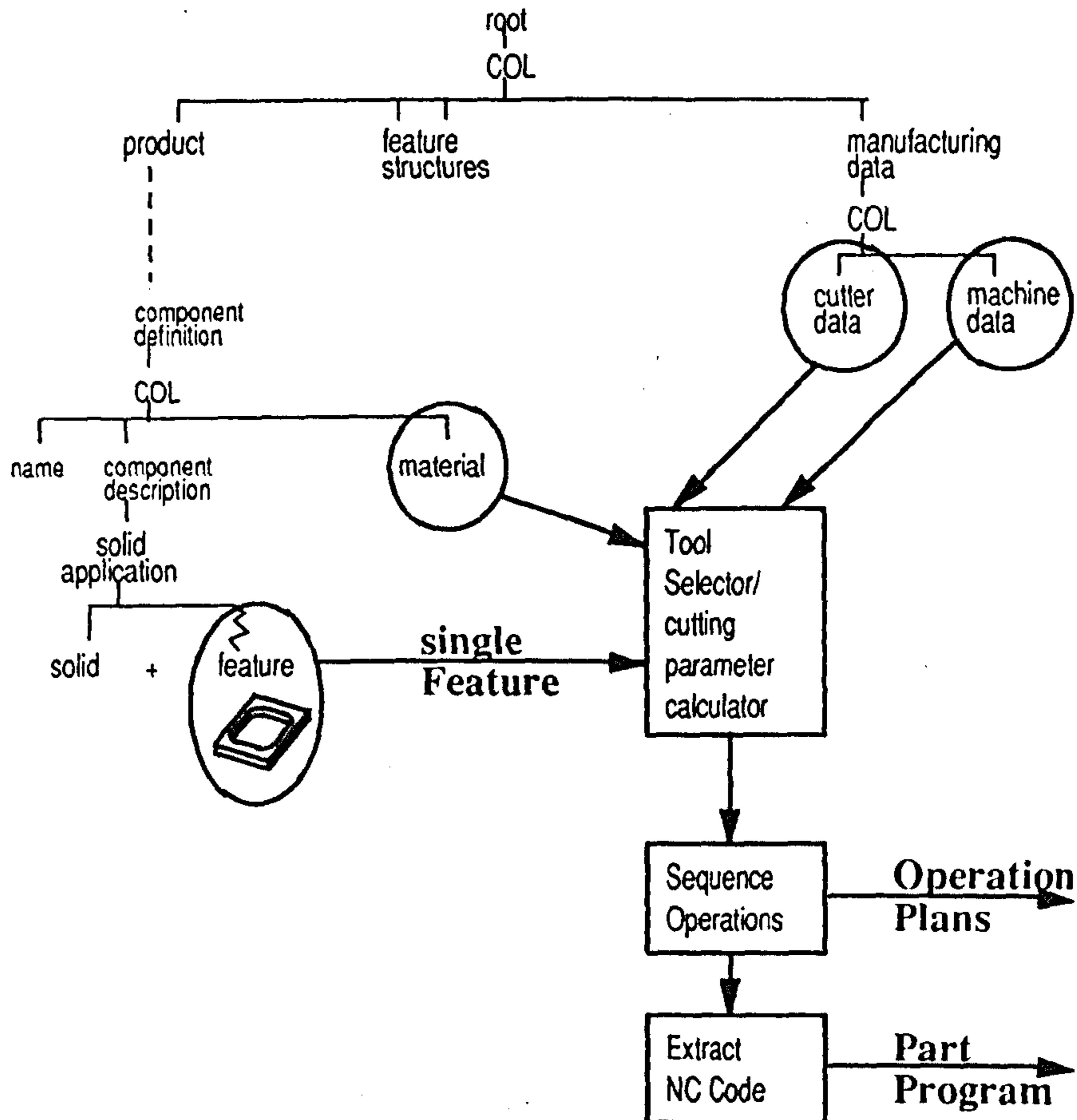
The experiments described in this section helped to define and confirm the feature level process data subset of the PDM and the planned processes subset of the PDM defined in chapter 6. They also demonstrate the success of the interaction between geometry and the Machine Planning Application. argued in chapter 7.

#### **11.3.1 Building and Testing a Set of Pre-defined Machining Features**

This set of experiments were aimed at checking the ability of the operation data sub-set of the PDM, defined in chapter 6.4, to provide appropriate data, from feature descriptions, to the Machine Planning Application. This was achieved by building a set of pre-defined machining features, as described in chapter 10.4.1; using each feature definition, in turn, to produce a component description using the Solid Application, described in chapter 10.3.3; generating machining instructions

and part programs using the Machine Planning Application, as illustrated in figure 11.3; and machining each feature, to check the validity of the machining code.

A set of features were defined in the feature library, covering a range of operation

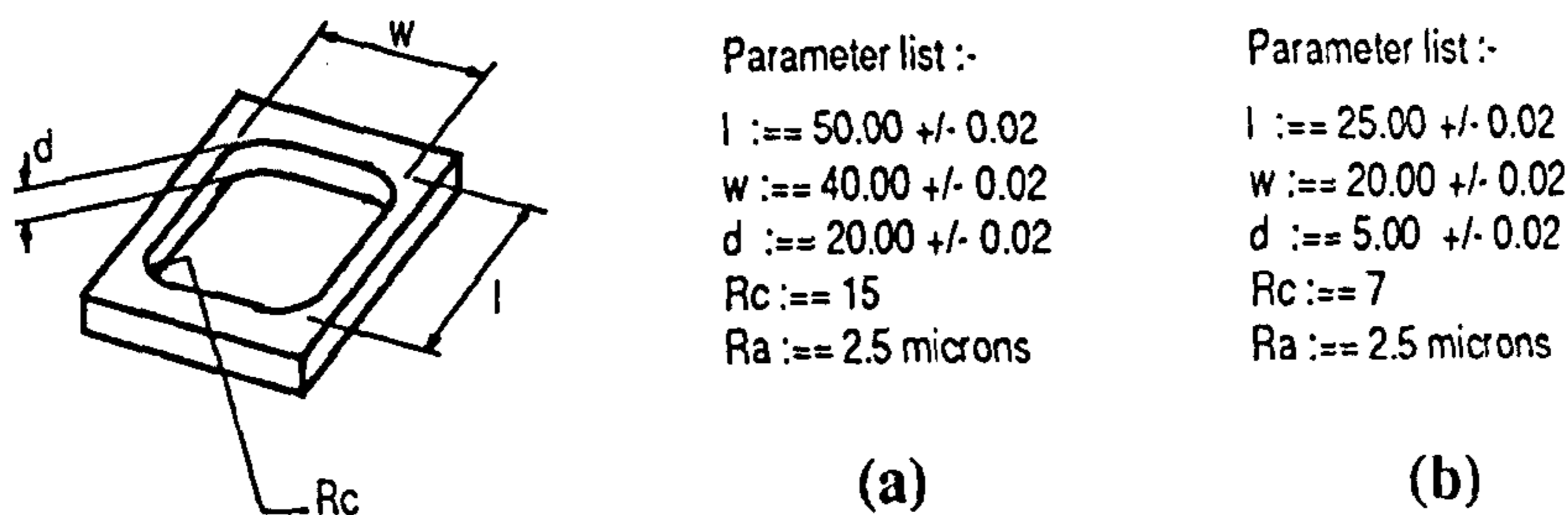


### The Use of the Feature Operation Data Structure to Generate Machining Information

Figure 11.3

types and also using the pre-defined tool path plans of Corrigan [36]. The features defined were a through pocket, a closed pocket, a slot, a channel, a hole, a four corner hole and a face feature as represented in appendix 4.

From these feature definitions, component descriptions were built, using each feature in turn, and used to generate operation plans and part programs. The resulting machining instructions and part programs were then successfully used to machine the defined features. An illustration of the parameters used for two instantiations of the closed pocket is provided in figure 11.4. The feature parameters used, and the machining instructions generated for specific features, are document-



**Parameter Values in Instances  
of the Closed Pocket Feature**

**Figure 11.4**

ed in appendix 7.2.

This experiment showed that pre-defined features could be built using the PDM framework and that the operation data sub-set of the PDM, representing a sequential set of operations, could provide the necessary data to the planning functions being used. At this stage in the exploration, the process data fields of the feature definition, other than operation data, had not been added. These were introduced for the setup planning experiments and were not required when addressing individual features or features confined to a single setup.

### **11.3.2 Sequencing Operations and Extracting NC Code for Interacting Features in a Setup.**

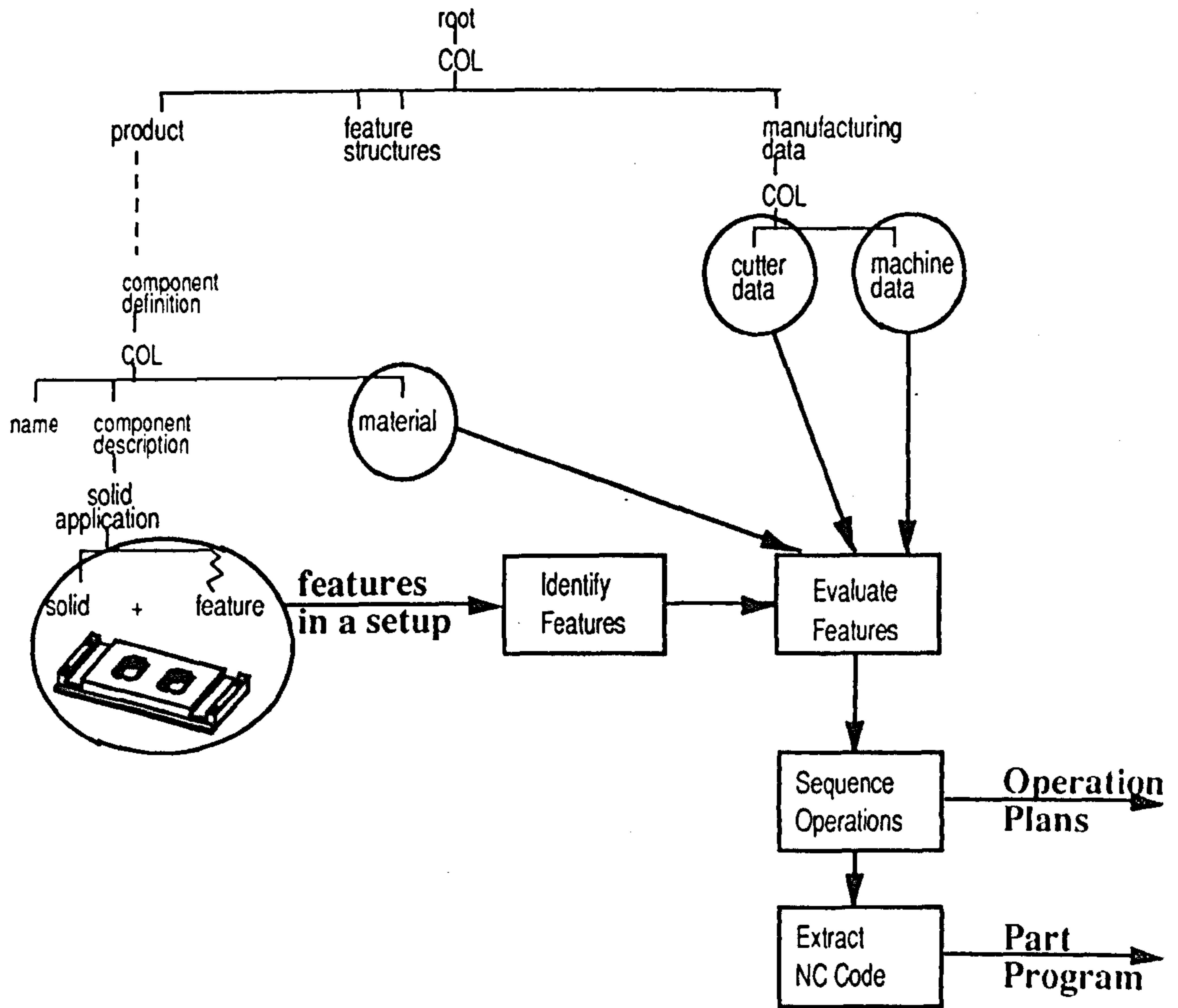
The experiments under this heading were used to show how features could be positioned within a component description, and how the resulting feature data could be



extracted, and evaluated, by the Machine Planning Application, resulting in operation plans and a part program, as illustrated in figure 11.5. The features which had been built and tested in the previous experiments were used to build variants of the Bolster Plate workpiece, illustrated in figure 11.6. Two experiments were performed, both using the same set of features, but using alternative dimensions. This introduces the check that the Closely Coupled Application could capture different dimensions, from the same feature definition, and generate appropriate machining instructions accordingly.

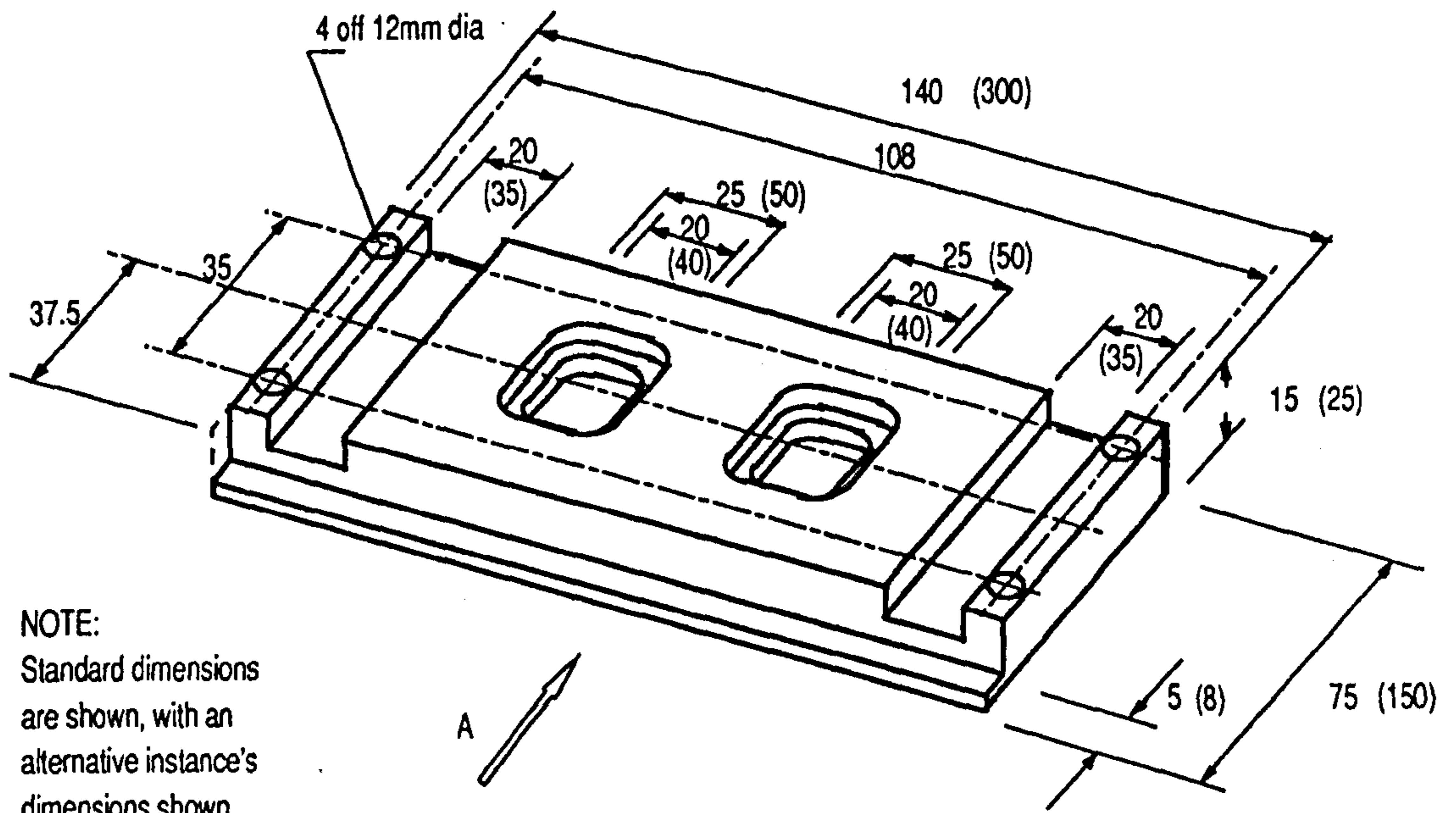
Component description for the Bolster Plate variants 1 and 2, shown in figure 11.6 (b) and 11.6 (c), were built, with the component description for the latter version documented in appendix 4.5.1. These component descriptions contain a range of Facing, End Milling, Closed and Open Slot Milling operation types which use similar tooling types in their manufacture. The features used provided a variety of tool path plans which were also to be brought together into a part program. It was assumed that a vice would be used for fixturing. The Machine Planner Application was linked to the PDM and used to generate machining instructions and a part program, as illustrated in figure 11.5.

It was found that simply applying features to the component model provided no basis for the identification of a fixture datum which could be used in part program generation. It was decided at this stage to assume a datum point at the front left top corner of the component and to position the global coordinate system at that point. This meant that the global transformation of each feature onto the component would reflect the position of each Operation with respect to the fixture datum, a requirement discussed in chapter 7.3.1. It was also assumed, that the overall stock dimensions would be the same as the component overall dimensions. The stock dimension and fixture datum problems were then reconsidered in relation to setup planning as described in the following section.

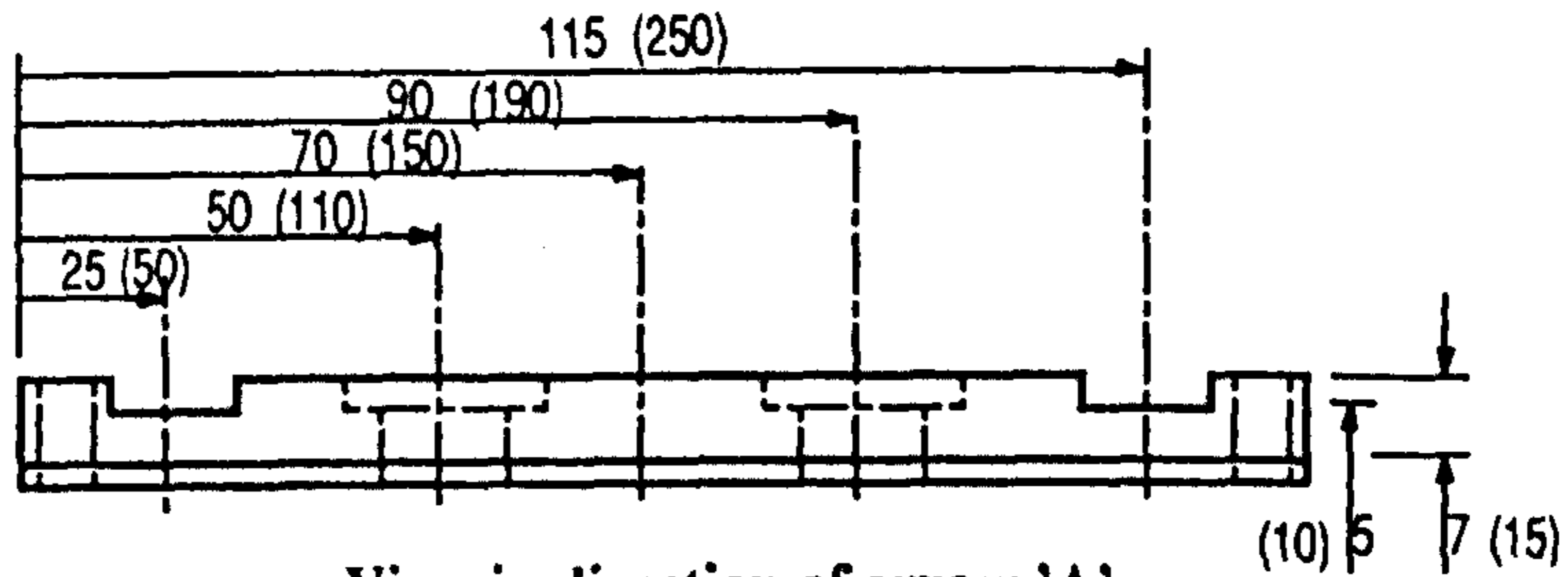


**The Generation of Machining Information from Interacting features in a Setup**

**Figure 11.5**

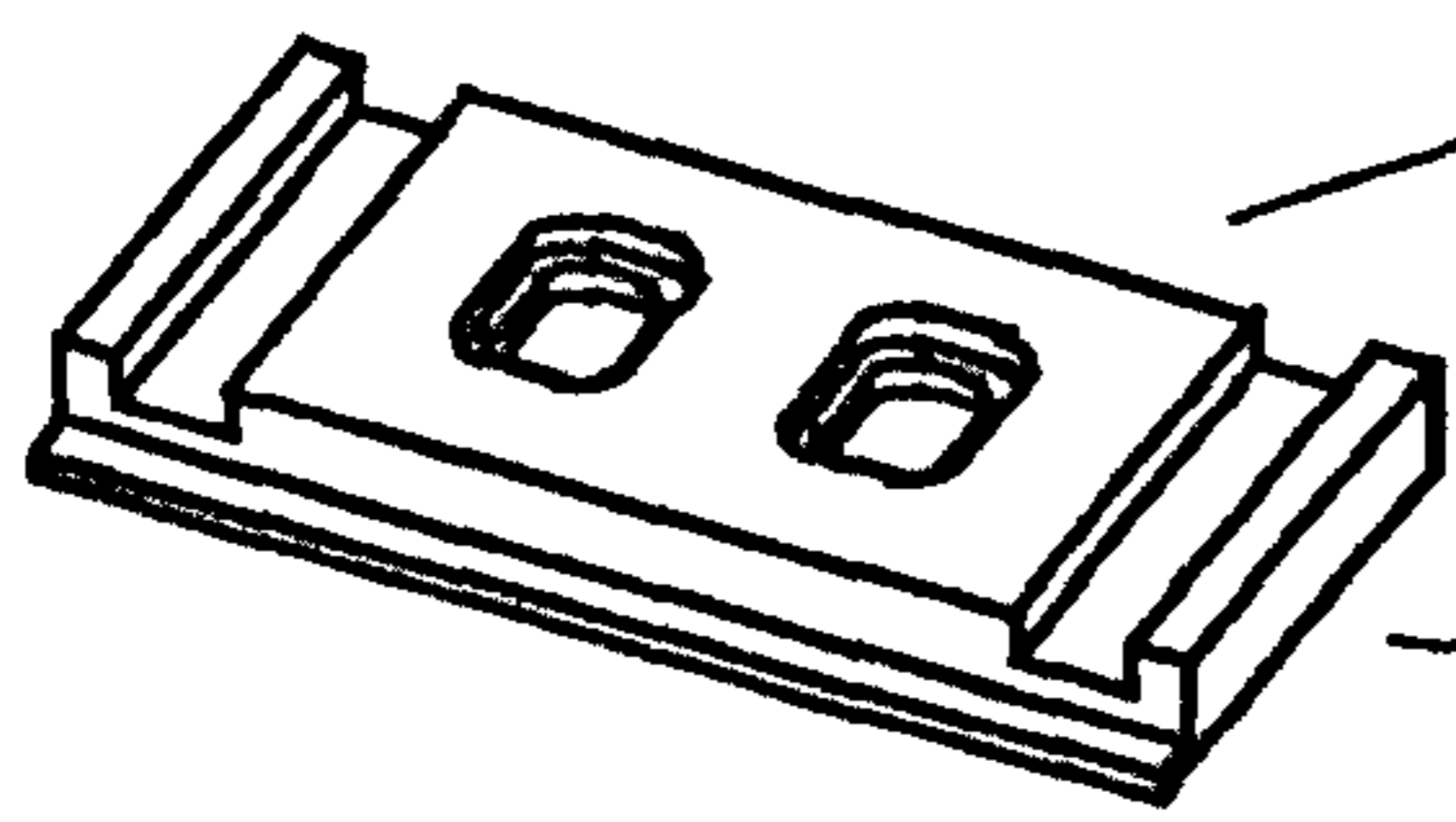


NOTE:  
Standard dimensions  
are shown, with an  
alternative instance's  
dimensions shown  
in brackets

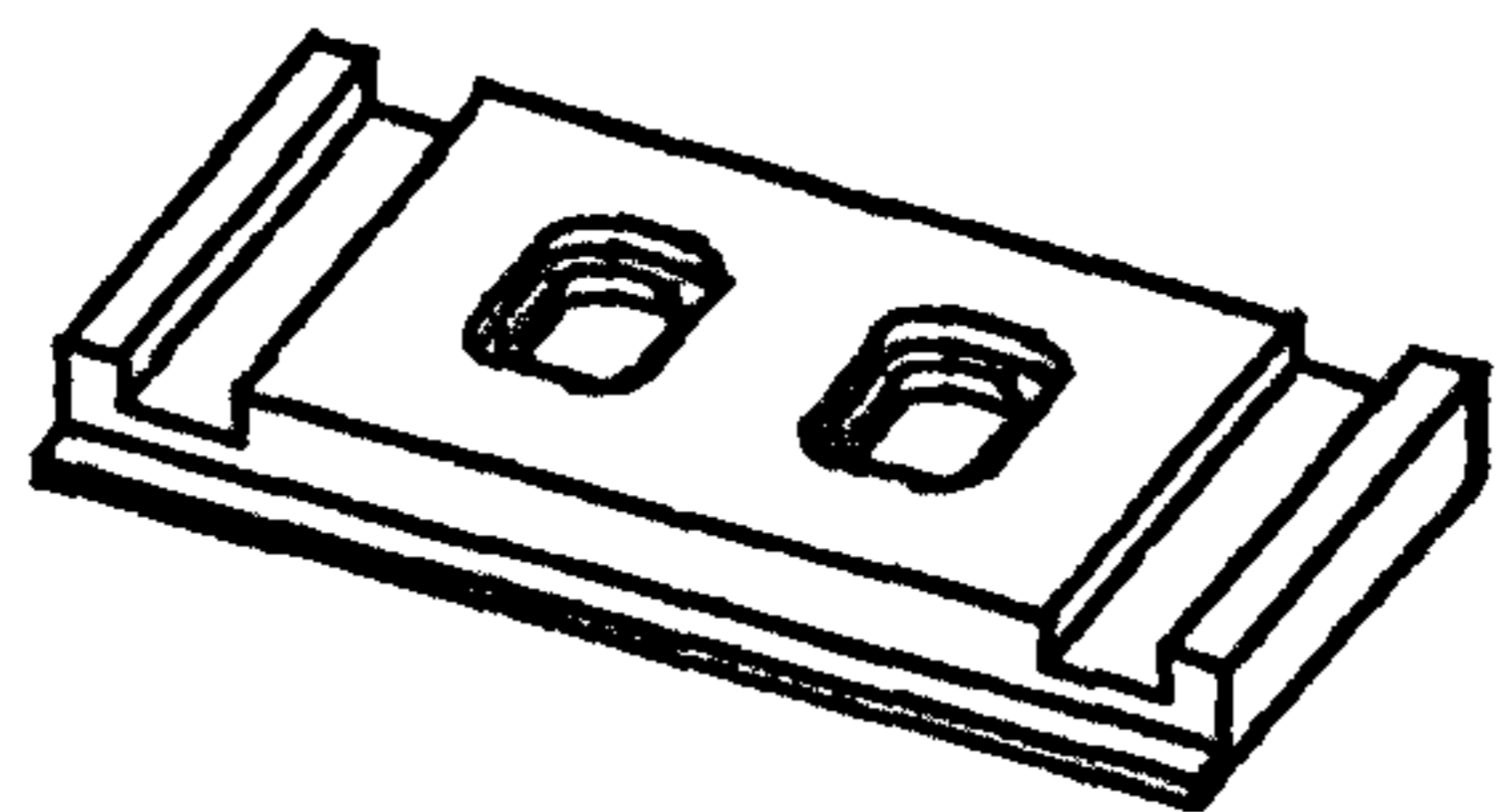


View in direction of arrow 'A'

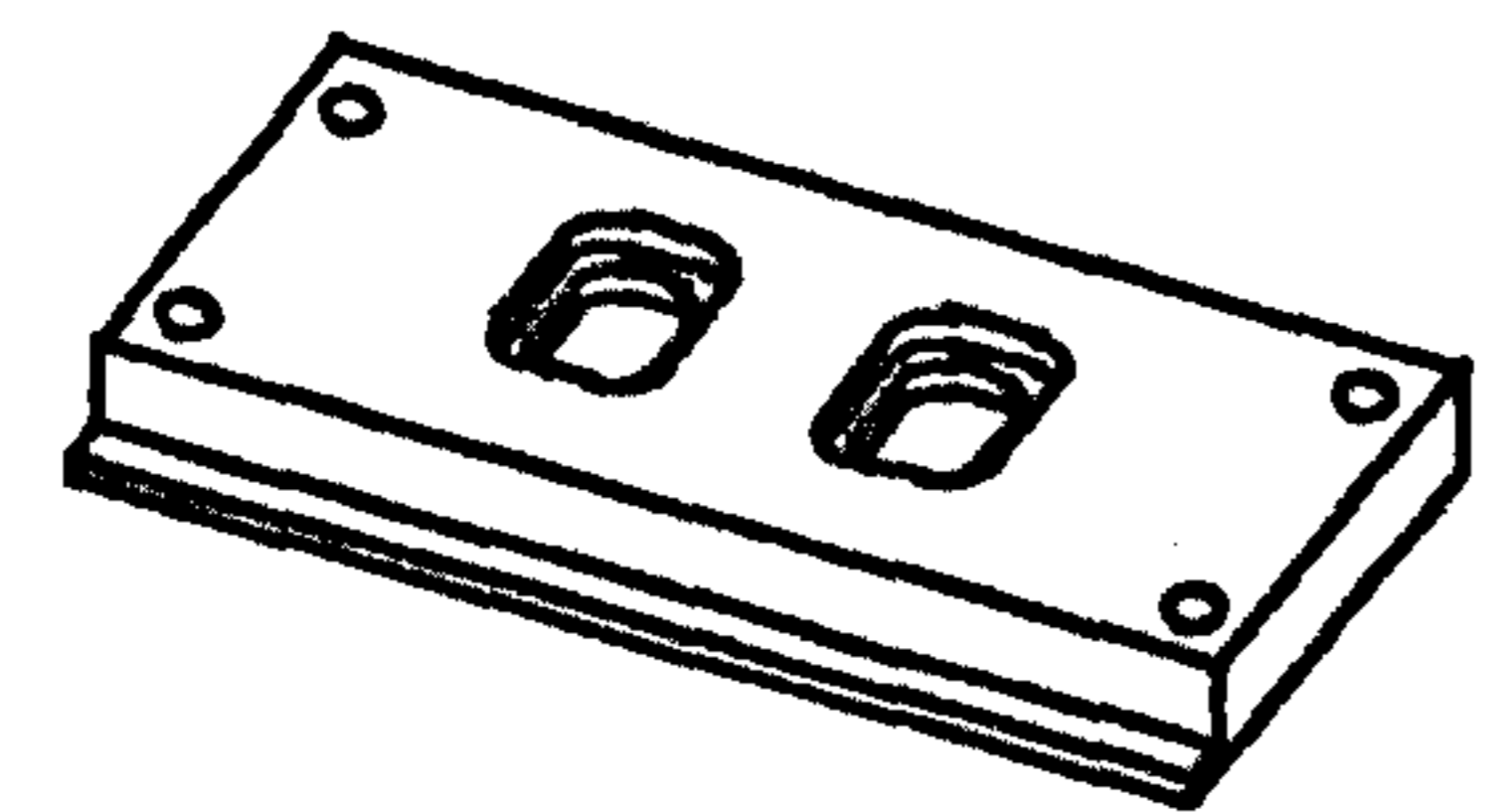
**An Instance of the Bolster Plate (a)**



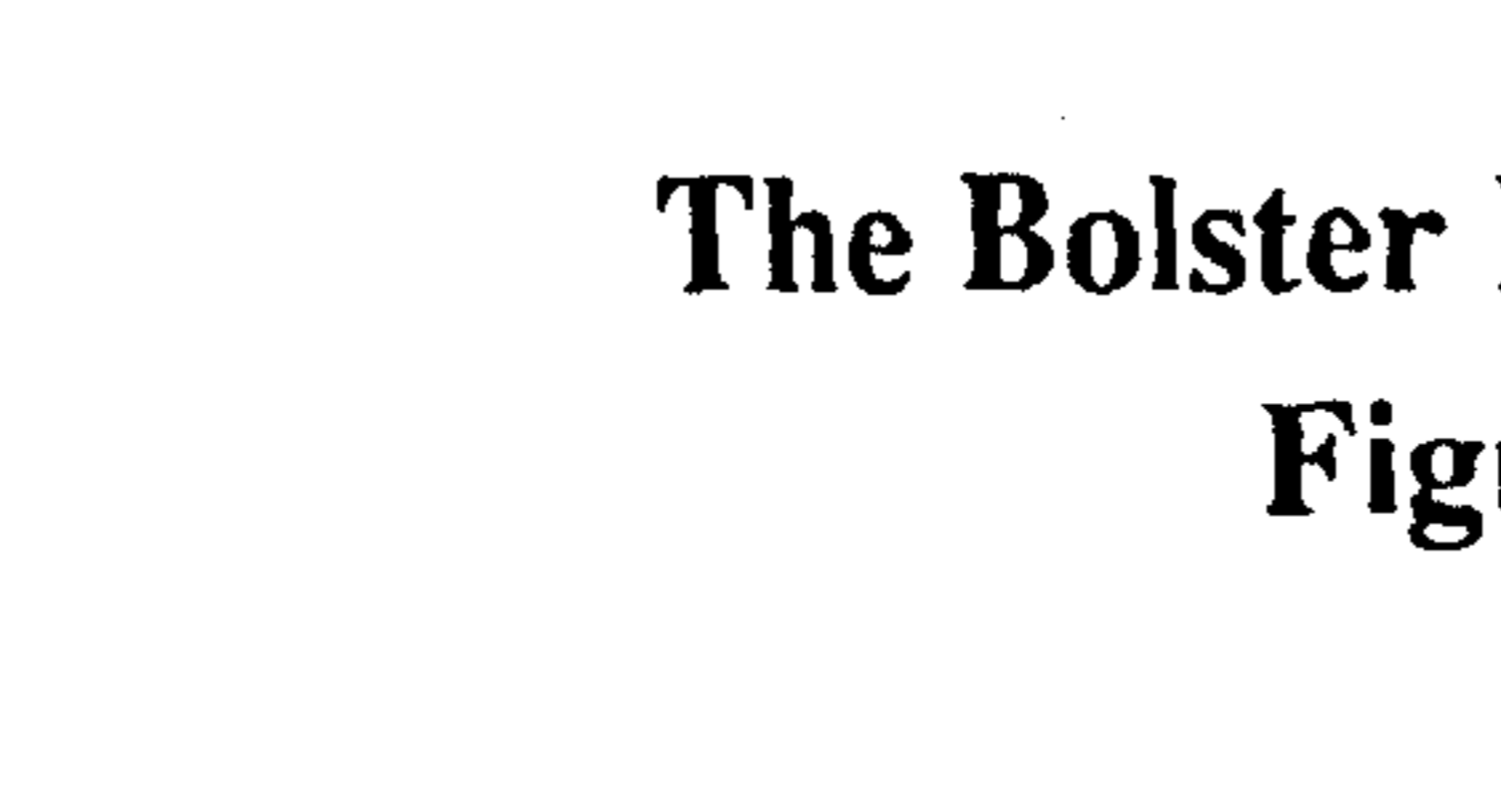
**(b) variant 1**  
component dimensions - (standard)  
stock dimensions - same as component length, width and depth



**(c) variant 2**  
component dimensions - (alternative)  
stock dimensions - same as component length, width and depth



**(d) variant 3**  
component dimensions - (standard)  
stock dimensions - L=150, W=100, D=25



**(e) variant 4**  
component dimensions - (standard)  
stock dimensions - L=150, W=100, D=25

**The Bolster Plate Workpiece  
Figure 11.6**



The experiments showed that Operation sequences and part programs could be generated, and successfully used to machine components for both component descriptions. Listings of the experimental output are not included in appendix 7, as output from the experiments described in sections 11.3.4 and 11.4 are listed, and these provide operation plan and part program outputs, as a part of the total machine plan.

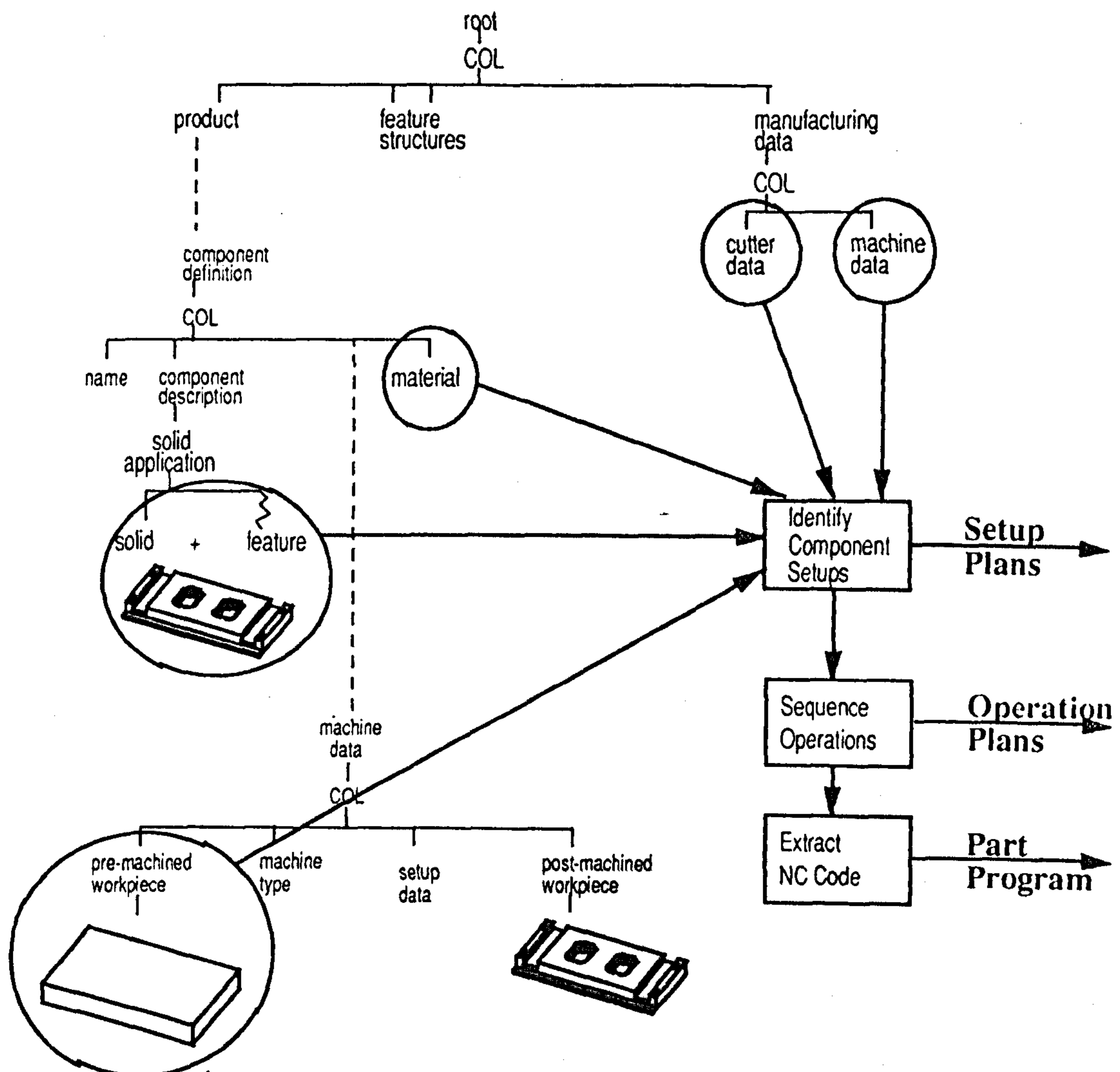
A number of factors were unresolved in this experiment. These were feature precedence identification; inter-feature dimensional constraint identification; an appropriate method for the identification of datum positions and the selection of appropriate fixtures.

Inter-feature tolerance implications for planning were not be included in the implementation. Features were positioned on the component by the use of geometric transformations relative to the global coordinate system. These transformations did not offer an appropriate mechanism for the representation of inter-feature dimensional constraints and no other mechanism was identified. The Relationship Graph was not available at this point in the work.

Through the work of P. Bell [12], it was proposed to generate a geometric proximity relationship between features which could satisfy the need for feature precedence data in operation sequencing. The author, therefore, pursued the problem of capturing the product data required for the integration of setup planning in the Machine Planning Application.

### 11.3.3 The Generation of Setup Planning data from Features which Interact across Multiple Setups.

The experiment described in this section was performed to explore the feature process data sub-set of the PDM, described in chapter 6.4, and the use of general component and stock geometry in the generation of setup planning data. The use of the Side Clamping fixturing strategy, described in chapter 9.2.3, was assumed and no geometric interaction checks were made. The links made from the PDM to



**The Generation of Machining Information from Interacting features in a Multiple Setup Workpiece**

**Figure 11.7**

the Machine Planner are illustrated in figure 11.7. The Bolster Plate variant 3 was

used, as illustrated in figure 11.6(d) which required all sides of the workpiece to be machined.

The features in the feature library were re-defined to contain the feature process data sub-set described in chapter 6.4. A component description of Bolster Plate variant 3 was built in the PDM, using Face features to achieve the link from the component to the stock dimensions, as explained in chapter 10.4.4. Stock dimensional parameters, in addition to the component dimensional parameters, were defined as general parameters in the feature library, and captured in the component level planned processes structure described in chapter 6.3. This structure was not fully defined, in this experiment, but contained fields for the pre and post machined workpiece description, providing the required input to the Machine Planner.

Machining instructions and part programs were generated successfully from the use of the Closely Coupled Machine Planner. These are not included in appendix 7, as the results of the experiments described in the next section capture similar data, of which the machining instruction for the Side Clamping Strategy are documented in appendix 7.3.3

As a result of this experiment the feature level process data sub-set of the PDM was shown to provide the necessary feature specific planning data inputs to the Machine Planner. The following points concerning the general geometric data manipulation could also be made:

- The conversion of Face features to profiles, in the process of using a fixturing strategy as argued in chapter 7.3.2, was used successfully.
- The use of component and setup coordinate axes, argued in chapter 7.3, provided the means for the identification of the relative positions of machining features.
- The representation of datum features as a part of a fixturing strategy has provided an appropriate mechanism to capture fixture datum features.

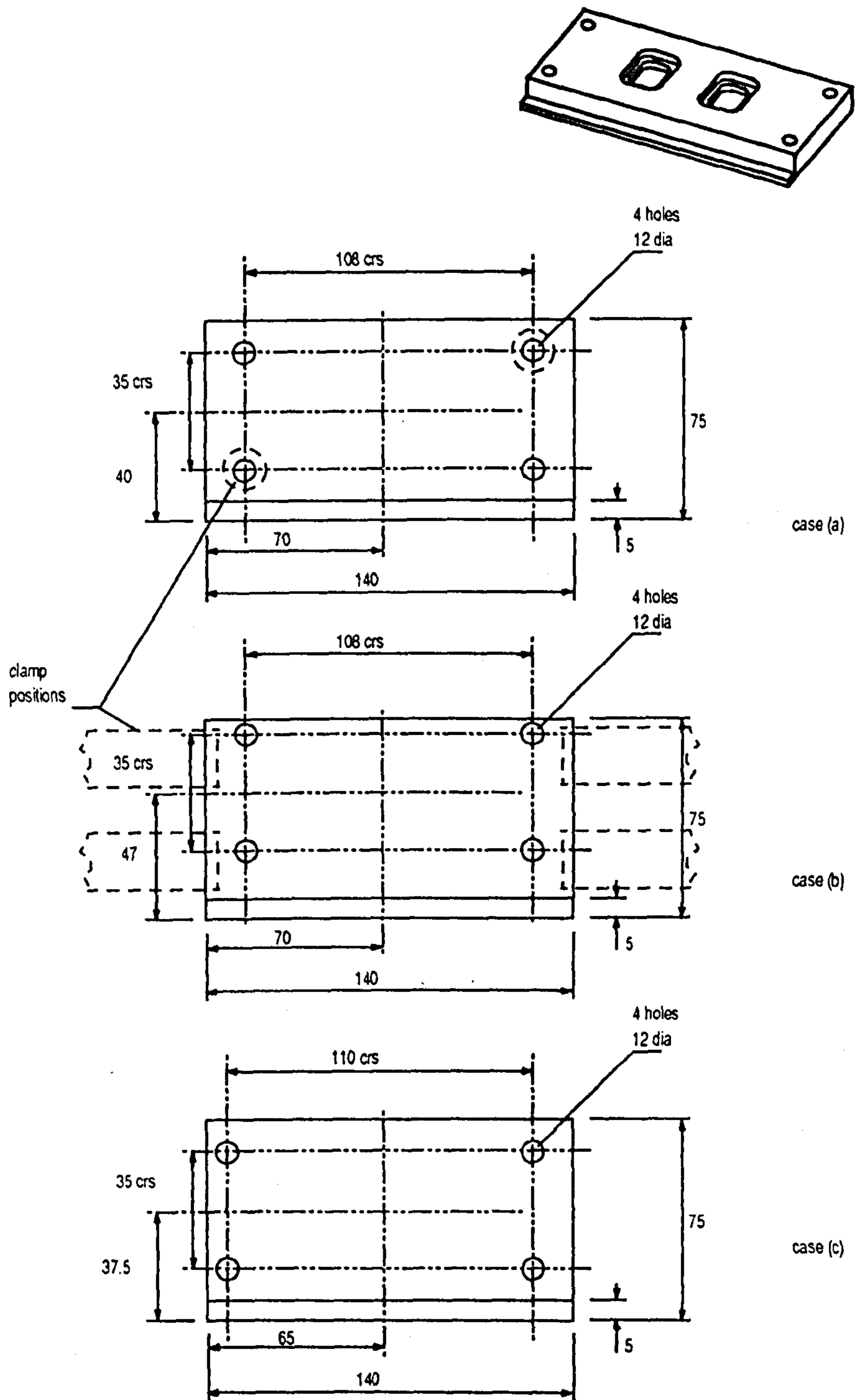


- The method of assessing the machined state of a datum feature, as described in chapter 7.3.1, was successful in capturing the position of the datum relative to the setup coordinate axe.

### 11.3.4 Fixture Strategy Selection

These experiments were performed to demonstrate how the geometric queries, argued in chapter 7.4, could be used to identify fixture and machining feature interactions, and hence influence fixturing strategy selection. In order to explore these interactions, a component description of variant 4 of the Bolster Plate workpiece, illustrated in figure 11.6 (e), was built, with a representation of the component description provided in appendix 4.5.2. The component description was used in conjunction with the full Machine Planner implementation described in chapter 9, including the three fixturing strategies of Side Clamping, Through Clamping and Down Clamping described in chapter 9.2.3.

To test the ability of the geometric queries to identify feature interactions, three sub-variants of the Bolster Plate variant 4 were described. These are illustrated in figure 11.8 and have been defined such that Case (a) should result in the use of the Through Clamping strategy; Case (b) should fail the Through Clamping strategy and result in the Down Clamping strategy; and case (c) should fail both the Through Clamping and the Down Clamping strategies and select the Side Clamping strategy. The Machine Planning software was used with each of the above sub-variants and machining instructions and part programs were generated. In each case the experimental results confirmed the expected results. The machining instructions generated for each are documented in appendix 7.3. This confirmed that a cell decomposition model was an appropriate model on which to base geometric queries to identify feature interaction data.

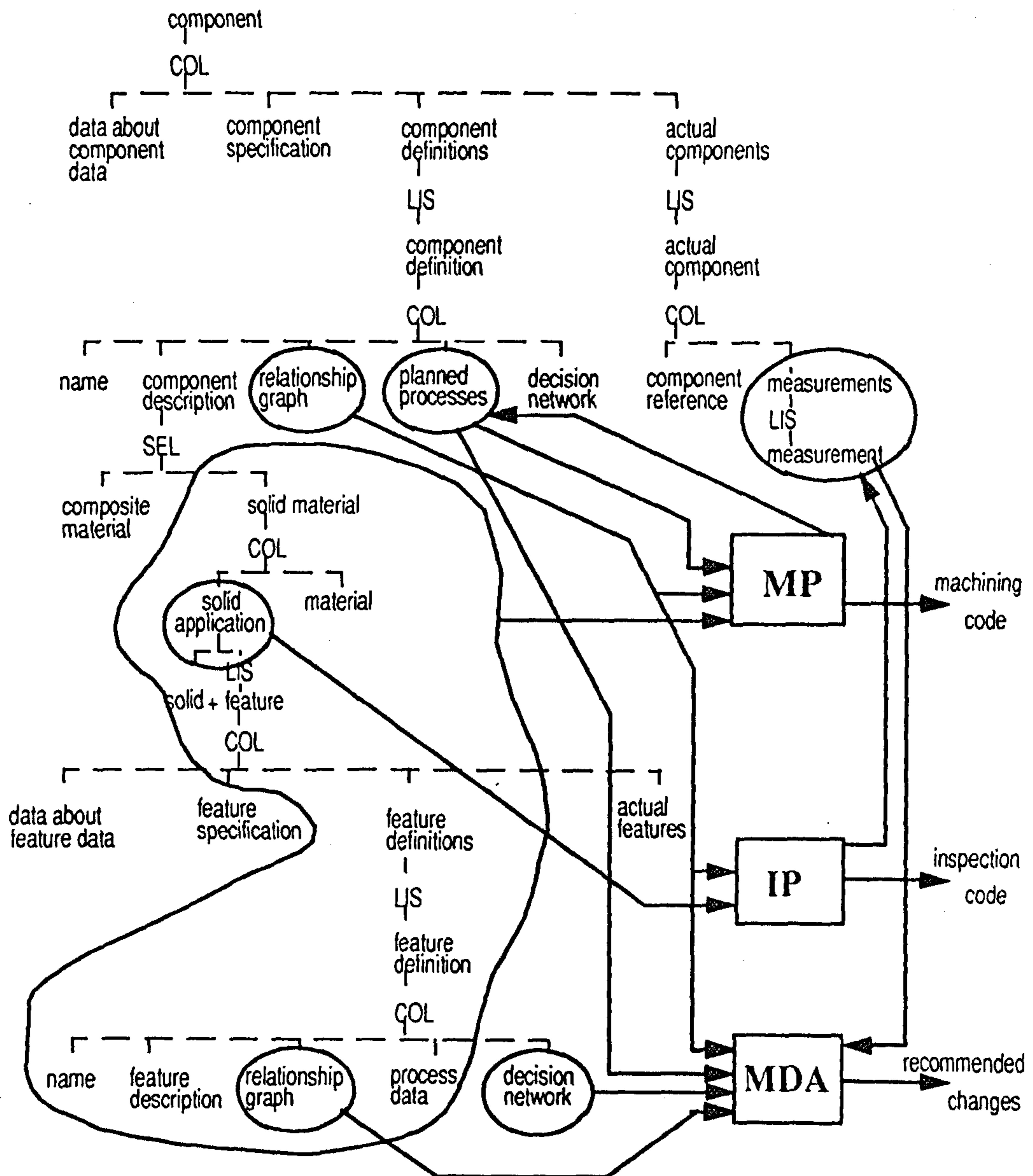


### Hole Position Variants to Test Feature Interaction Queries

Figure 11.8

### 11.4 THE INTEGRATION OF MANUFACTURING CODE GENERATION THROUGH THE PDM

This experiment brought together the product data requirements of the Manufacturing Code Generation Applications of Machine Planning, Inspection Planning and Manufacturing Data Analysis, to explore the viability of their integration, through



MCG Activity Integration in the PDM

Figure 11.9

the Product Model environment. This experiment involved all members of the ISS



research team and was aimed at demonstrating the level to which the integration of the work of the various team members could be achieved through the PDM. To perform this experiment the final version of the PDM, "pms\_8.lam", was constructed, a component description of the Glacier reduced size bearing built, and each of the three MCG Application implementations were linked to the PDM. The data flows expected between the Applications, via the PDM, is illustrated in figure 11.9, while a diagram of the Glacier Component used in the experiment is shown in

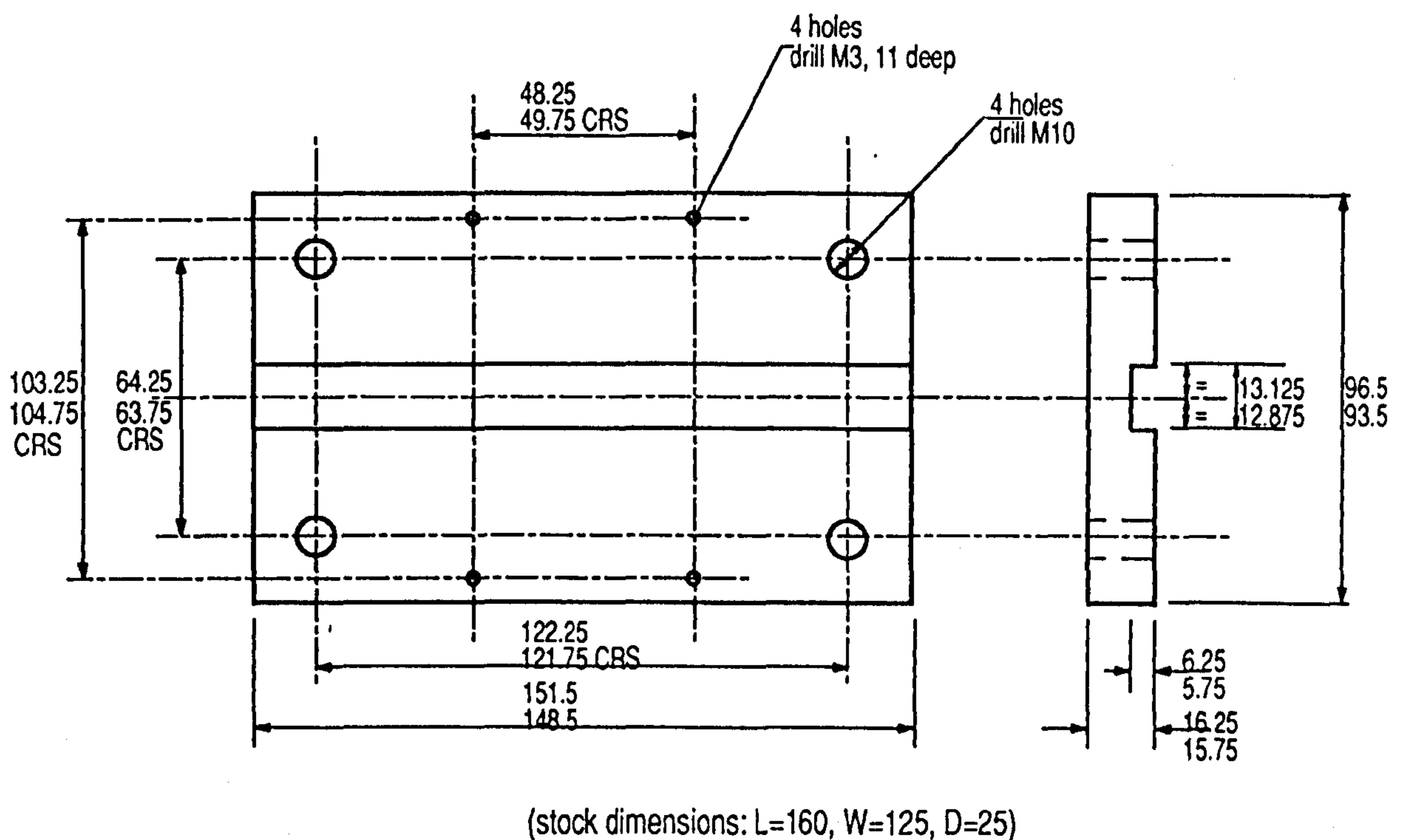


figure 11.10.

### The Glacier Reduced Size Bearing Figure 11.10

The experimental component description was built using Pre-defined features as in the previous cases. The Relationship Graph was now implemented, but this provided only links to geometric surfaces with no identifying links to feature data as discussed in chapter 8. The Machine Planning software was used to generate machining instructions and part programs for the component, following the procedure described in chapter 10.5.2. These machining instructions and part programs, listed in appendix 7.4, were used to produce a testpiece on the Wadkin V4/6 machining centre. Using the same Product Model, Plans and code to inspect the component

were produced by the Inspection Planner and used to check the machined component. The Machining Plans and resultant measurements were then used by the Manufacturing Data Analysis Application to check the inspected results. Each of these activities called upon the common base of product data in order to perform its function, hence demonstrating the integration ability of the Product Model environment.

### Postscript

Work on the parallel path of machining code generation through the analysis of complex geometries was the responsibility of P. Bell [12]. Bell concentrated on achieving computationally effective algorithms to achieve the automatic generation of NC code. As such his work had a different emphasis from the author's, but has the potential to offer solutions in areas related to geometries beyond the scope of this thesis.

The two levels of *Identify Features*, MP11, and *Evaluate Features*, MP13, in the IDEF0 model in appendix 3, were produced by the author in combination with P. Bell. These offer a model of how pre-defined features, and feature recognition techniques could be combined within Machine Planning. The work of Bell has gone some way to fulfilling the *Evaluate Features* requirement for the general machining case [158], breaking down the pre and post setup geometry to identify Operation data which can be used in the *Identify Cutting Tools*, MP1323 activity. His work uses the SDSM and concentrates on rough machining. There is still a need to explore the analysis of geometry to identify finish machining operations.

Bell has demonstrated his work within the planned processes data sub-set of the PDM described in chapter 6.5.3, [158]. This provides further evidence that this data sub-set of the PDM offers an appropriate basis for the capture of necessary planning information and hence offers encouragement that this can offer an integrating capability between the pre-defined feature route to capturing process data and feature recognition routes requiring complex geometric analysis.

**CHAPTER 12****CONCLUSIONS AND RECOMMENDATIONS FOR FURTHER WORK****12.1 INTRODUCTION.**

The research described in this thesis has explored the Integration of a Machine Planner in a Product Model Environment. An experimental Machine Planner has been implemented to test the thesis that a Closely Coupled Machine Planner can be integrated, through the Product Data Model which has been described in the research. The experiments which have been performed to explore the thesis have led to the conclusions and recommendations for further work, which are made in this chapter.

**12.2 CONCLUSIONS**

1. It has been shown that a Machine Planner can function as a Closely Coupled Application in a Product Model Environment. This has been demonstrated by building and populating a PDM, implementing the experimental Machine Planning Application, and integrating these to generate machining instructions and part programs which have been used to machine testpiece components.

2. It has been demonstrated, through collaboration with other researchers, that a range of design and manufacturing Applications can be integrated through the use of a PDM. Enough evidence has been gathered to suggest that a Closely Coupled Machine Planner offers a potentially more powerful capability than a stand alone



system, because of the interactions which are made possible. In particular, links to comprehensive geometric and tolerance analysis routines and to data feedback systems offer this potential.

3. The understanding of Product Modelling has been advanced, by defining the lower levels of a Product Data Model, necessary for the integration of a Machine Planner. The feature level process data sub-set and the component level planned processes sub-set of the Product Data Model have been defined.

4. The lower levels of the Product Data Model, which have been defined, have been shown to capture necessary product data for Machine Planning. Product Data Models have been populated, and used to provide data to a Closely Coupled Machine Planner.

5. It has been shown how a feature based component description can be analysed to generate machining part programs. In particular the author's work has explored relationships between global, component, setup and feature coordinate systems and has demonstrated, through the use of fixturing strategies, how fixture datum points can be identified, providing a reference point for part program generation.

6. Feature interactions can be identified through the use of geometric queries on a cell decomposition model. In particular, interactions between fixture positions and machining features have been identified in the experimental implementation.

### 12.3 RECOMMENDATIONS FOR FURTHER WORK

1. There is a requirement to prove that geometric queries on the SDSM are an appropriate mechanism to check for thin wall problems. To show this, a similar approach to that described in chapter 9.3, should be followed. However, the cell size used should be smaller than that used in the fixture and machining feature interaction queries which have been implemented, to ensure that the accuracy of query is adequate.

2. In future work on integrated systems the relationship between features, dimensions and tolerances and Machine Planning should be a major priority. The method described in chapter 10, linking from the Relationship Graph to feature descriptions should be explored. Also tolerance analysis methods, to compare design requirements with expected manufacturing tolerances, should be investigated.

3. A flexible representation of machine capability is needed in future integrated systems. In an integrated design and manufacture environment machine capability information should be available to a range of Applications. For example, in the ISS programme, the MDA work of L. Lee [82], would have been able to influence the code generation of a Machine Planner, if it could have accessed the representation of fixturing strategies and machining rules.

4. There is a need to explore more complex geometries, and how feature recognition techniques can provide the breakdown of such geometry into the meaningful data required for planning. Although feature recognition has been outside the scope of this work, the feature level process data sub-set of the PDM, and the component level planned processes sub-set of the PDM, should provide guidance to future research work on the data such recognition systems should be expected to generate.



## REFERENCES

1. Adiga, S. and R.K. Li, "Research in Application of Artificial Intelligence Techniques to Generative Process Planning in Manufacturing," Proc. IFIP Conference on Advances in Production Management Systems, North Holland, 1987.
2. Alder, G.M., J.A. McGeogh, and K.K.B. Hon, "Selection of Machining Processes by Intellegent Knowledge Based Systems," International Conference on Computer-aided Production Engineering, 1986.
3. Alder, R.E. and K. Ishii, "DAISIE: Designer's Aid for Simultaneous Engineering," Proceedings of Computers in Engineering Conference, ASME, 1989.
4. Allen, D.K. and P.R. Smith, "Computer Aided Process Planning," Computer Aided Manufacturing Laboratory, Brigham Young University, Utah.
5. Alting, L. and H. Zhang, "Computer Aided Process Planning: the State of the Art Survey," International Journal of Production Research, vol. 27, no4, 1989.
6. Applicon,, Bravo 3, NC General Reference Manual, Applicon, 4251 Plymouth road, PO Box 986, Ann Arbor, Michigan , 1987.
7. Armstrong, G.T., C.G. Carey, and A. De Pennington, "Numerical Code Generation from a Geometric Modeling System," Solid Modeling by Computers, Plenum Press, 1984.
8. Barkocy, B. and W. Zdeblick, "A Knowledge Based System for Machine Operation Planning," The Carbide and Tool Journal, July/August 1985.
9. Beigel, J., "The Future Role of Expert Systems in Manufacturing," Proc. 8th. annual conf. on Computers and Industrial Engineering, 1986.
10. Bell, P., "User14 - The Structure Editor Menu Option - Machining," GMP Project, Dept. of Mechanical Engineering, University of Leeds, 1987.
11. Bell, P.K., "Automatic Cutter Path Generation.," Dept of Mechanical Engineering, Leeds University, 1987.



12. Bell, P.K., "General Region and Operation Determination," Iss-msg-report-15, Department of Mechanical Engineering, Leeds University, 1990.
13. Bell, R. and A. De Pennington, "Information Support Systems for Design and Manufacture Project," Research Proposal, Dept. of Manufacturing Engineering, Loughborough University of Technology, 1987.
14. Bell, R. and R.I.M. Young, "Machine Planning: its Role in the Generation of Manufacturing Code from Solid Model Descriptions," International Journal of Production Research, vol. 27, no5, 1989.
15. Bell, R. and A De Pennington, "Information Support Systems for Design and Manufacture - Final ACME report," Dept of Manufacturing Engineering, Loughborough University of Technology, 1990.
16. Bloor, S.M., A De Pennington, S.B. Harris, D. Holdsworth, A. Mckay, and N.K. Shaw, "Towards Integrated Design and Manufacturing," Factory 2000 conference, 1988.
17. Boerma, J.R. and J.J.Kals, "FIXES, a System for Automatic Selection of Set-ups and Design of Fixtures," Annals of the CIRP, 1988.
18. Boerma, J.R. and J.J.Kals, "Fixture Design with FIXES: The Automatic Selection of Positioning, Clamping and Support Features for Prismatic Parts," Annals of the CIRP, 1989.
19. Boothroyd, G. and W.A. Knight, Fundamentals of Metal Machining and Machine Tools, McGraw-Hill, 1989.
20. Brimson, J. and P.J. Downey, "Feature Technology: A Key to Manufacturing Integration," CIM Review, spring 1986.
21. C-Plan,, "Computer Aided Process Planning," CAD Centre, Cambridge, England CB3 0HB, 1985.
22. Camtek,, Pepsm2 Milling Manual, version 2.2, Camtek Ltd, 1988.
23. Carey, C.G., PhD Thesis, Dept of Mechanical Engineering, Leeds University, 1987.
24. Carlier, J. and J. Peters, "MOPS - A Machining Centre Operation Planning System," Annals of CIRP, vol. 34, no. 1, 1985.

25. Chan, S.C. and H.B. Voelcker, "An introduction to MPL - a new machining process/programming language," Proceedings IEEE Conference on Robotics and Automation., 1986.
26. Chang, T.C. and R.A. Wysk, "An Integrated CAD / Automated Process Planning System," AIIE Transactions, Sept. 1981.
27. Chang, T.C. and R.A. Wysk, An Introduction to Automated Process Planning Systems, Prentice Hall International, 1985.
28. Chang, T.C., D.C. Anderson, and O.R. Mitchell, "QTC - An Integrated Design/Manufacturing/Inspection System for Prismatic Parts.," Purdue Engineering Research Centre for Intelligent Manufacturing Systems, 1988.
29. Chen, S.J., S. Hinduja, and G. Barrow, "Automatic Tool Selection for Rough Turning Operations," International Journal of Machine Tools and Manufacturing, Vol.29 No.4 1989.
30. Choi, B.K., M.M. Barash, and D.C. Anderson, "Automated Recognition of Machined Surfaces from a 3D Solid Model," Computer Aided Design, vol. 16, no 2, March 1984.
31. Choi, B.K. and M.M. Barash, "STOPP: an Approach to CAD/CAM Integration," Computer Aided Design, vol. 17, no 4, May 1985.
32. Chou, Y.C. and M.M. Barash, "Computerized Fixture Design from Solid Models of Workpieces," Integrated and Intelligent Manufacturing, the annual meeting of the ASME, 1986.
33. Chung, J.C.H., R.L. Cook, D. Patel, and M.K. Simmons, "Feature Based Geometry Construction for Geometric Reasoning," Proceeding of Computers in Engineering Conference, ASME, 1988.
34. Colquhoun, G.J., J.D. Gamble, and R.W. Baines, "The Use of Idef0 to Link Design and Manufacture in a CIM Environment," International Journal of Operation and Production Management, 1989, vol.9. No.8..
35. Corrigan, M.J., "Inspection Plan and Code Generation for Coordinate Measuring Machines in a Product Modelling Environment," PhD Thesis, Dept of Manufacturing Engineering, Loughborough University of Technology, 1990.



36. Corrigall, M.J., "Feature Machining using Parametrized Macros," Internal report ISS-1-DD-4, Dept. of Manufacturing Engineering, Loughborough University, 1988.
37. Cutkosky, M.R. and J.M. Tenenbaum, "CAD/CAM Integration Through Concurrent Process and Product Design," Integrated and Intelligent Manufacturing, ASME, 1988.
38. Cutkosky, M.R., J.M. Tenenbaum, and D. Muller, "Features in Process Based Design," Proceeding of Computers in Engineering Conference, ASME, 1988.
39. Cutkosky, M.R. and S.H. Lee, "Fixture Planning with Friction for Concurrent Product/Process Design," NFS Engineering Design Research Conference, 1989.
40. Darvishi, A.R. and K.F. Gill, "Knowledge Representation Database for the Development of a Fixture Design Expert System," Proceeding of the Institute of Mechanical Engineers, vol 202, no. B1, 1988.
41. Davies, B.J. and I.L. Darbyshire, "The Use of Expert Systems in Process Planning," Annals of CIRP, vol. 33/1/84, 1984.
42. Descotte, Y. and J.C. Latombe, "GARI: A Problem Solver that Plans how to Machine Mechanical Parts," Proc. 7th. International Joint Conference on Artificial Intelligence, 1981.
43. Desousa, R, "Tool Management in Highly Automated Batch Manufacturing Systems," PhD Thesis, Dept of Manufacturing Engineering, Loughborough University of Technology, 1988.
44. Detollenaere, M., W. DeWolf, H. Van Acker, J. Pinte, and Y.D. Willems, "ESOP An Expert System for Operation Planning on Machining Centres," CAM-I M-88-PPP-02, June 1988.
45. De Pennington, A., R. Woodhead, and Z. Dobolyi, "Process Planning as an Application for Expert Systems Technology," Proc. 8th. annual conf. on Computers and Industrial Engineering, 1986.
46. De Pennington, A. and R. Bell, "Integrated Design and Manufacture," Seminar, Dept. of Manufacturing Engineering, Loughborough University of Technology, March, 1990.
47. Ferreira, P.M., S.C-Y. Lu, and X. Zhu, "Conceptual Model for Process Planning," CAM -I, 1990.



48. Frost, R.A, Introduction to Knowledge based Systems, Collins Professional and Technical Books, 1986.
49. Furth, B., "Automated Process Planning," CAM-I M-87-PP-06 , Oct. 1987.
50. General, Electric, "Mark Century 2000 CNC programming Manual," General Electric, 1985.
51. Genord, D.C., M.G. Welch, R.V. Houser, and L. Wyttenbach, "An Automatic NC Processor: Harnessing the Technology of Form Feature Based Solids Modelling," Proceedings of Autofact , 1988.
52. Gindy, N., "A Hierarchical Structure for Form Features.," International Journal of Production Research, 1989.
53. Giusti, F., M. Santochi, and G. Dini, "COATS: an Expert System Module for Optimal Tool Selection," Annals of CIRP, 1986.
54. Grayer, A.R., "The Automatic Production of Machined Components Starting from a Stored Geometric Description," Advances in Computer-Aided Manufacturing, North Holland, 1977.
55. Groover, M.P. and E.W. Zimmers, CAD/CAM: Computer-Aided Design and Manufacturing, Prentice/Hall International, 1984.
56. Halvei, G. and R. Weill, "Influence of Manufacturing Tolerances on Fixturing of Machined Parts in Process Planning Systems," First CIRP Working Seminar on CAPP, 1985.
57. Ham, I. and S. Lu, "Computer Aided Process Planning: The Present and the Future," Annals of CIRP, Vol.37 1988.
58. Harmon, P. and D. King, "Expert Systems," J. wiley, 1985.
59. Henderson, M.R., "Extraction of Feature Information from 3 Dimensional CAD Data," Doctoral Thesis, Purdue University, 1984.
60. Hinde, C.J. and N.G. Alton, "Process Planning using a Truth Maintenance System," Proc. 2nd. UMIST/ACME Workshop on Advanced Research in Computer Aided Manufacturing, 1987.
61. Hinde, C.J., A.D. Bray, P.J. Herbert, V.A. Lauenders, and D. Round, "An Integrated Approach to Process

- Planning," International CIS Journal, 1989.
62. Hinduja, S., D.J. Petty, M. Tester, and G. Barrow, "Calculation of Optimum Cutting Conditions for Turning Operations," Proceedings Institute of Mechanical Engineers, vol 199, 1985.
  63. Hinduja, S. and H. Huang, "Automatic Determination of Work Holding Parameters for Turned Components," Proceedings of the Institute of Mechanical Engineers, Vol 203, 1989.
  64. Hummel, K.E. and S.L. Brooks, "Symbolic Representation of Manufacturing Features for Automated Process Planning Systems," ASME Knowledge Based Expert Systems for Manufacture PED-vol 24, 1986.
  65. Hummel, K.E., "Allied Bendix X-Cut System," Computer-Aided Manufacturing International M-87-PPP-02, 1987.
  66. Ichbiah, J.D., Reference Manual for the ADA Programming Language, Castle House Publications, ISBN 07194 0097X.
  67. Imamura, S., T. Kojima, H. Sekiguchi, and K. Inoue, "A Study on the Object Oriented Product Model - Representation of Geometry and Dimensions," Annals of CIRP, Vol 37/1, 1988.
  68. IMPACT,, "Integrated Modelling of Products and Processes using Advanced Computer Technologies," ESPRIT II Project No 2165, Proceedings of Berlin Workshop, 1991.
  69. Iwata, K. and N Sugimura, "Development of Product Model for Design and Manufacturing of Machine Products," North American Metal Research Conference 14th Proceedings (NAMRC), 1986.
  70. Iwata, K. and Y. Fukuda, "Representation of Know-How and its Application of Machining Reference Surface in Computer Aided Process Planning," Annals of CIRP, 1986.
  71. Jared, G., "Feature Recognition - The Latest from the Battlefield," Cranfield Institute, Luton, Bedfordshire, 1990.
  72. Jared, G.E.M., "Feature Recognition and Expert Systems for Operation Planning in NC Machining," Proc. International Conference AI Europa, 1986.



73. Joshi, S., N.N. Vissa, and T.C. Chang, "Expert Process Planning System with Solid Model Interface," International Journal of Production Research, vol. 26 no. 5, 1988.
74. Kahles, J.F., "Machinability Data Requirements for Advanced Machining Systems," Annals of CIRP, 1987.
75. Kanai, S., F. Tanaka, T. Amano, T. Kishinami, and K. Saito, "Heirarchical CAM System based on Cell-Constructed-Geometric-Model," Bulletin of Japan Society of Precision Engineering, June 1987.
76. Kanumury, M., J. Shah, and T.C. Chang, "An Automatic Process Planning System for QTC - An Integrated CAD/CAM System," Purdue Engineering Research Centre for Intelligent Manufacturing Systems, 1988.
77. Karinithi, R.R. and D.S. Nau, "Geometric Reasoning as a Guide to Process Planning," Proceeding of Computers in Engineering Conference, ASME, 1989.
78. Kimura, F., Y. Yamaguchi, Y. Sasaki, K. Kido, and M. Hosaka, "Construction and Uses of an Engineering Data Base in Design and Manufacturing Environments," File Structures and Data Bases for CAD, IFIP, North Holland, 1982.
79. Kimura, F., H. Suzuki, and L. Wingard, "A uniform approach to Dimensioning and Tolerancing in Product Modelling," Computer Applications in Production and Engineering, IFIP, 1987.
80. Koriba, M., "System modelling for the competetive edge," Chartered Mechanical Engineer, April 1988.
81. Kumara, S.R.T., S. Joshi, R.L. Kashyap, C.L. Moodie, and T.C. Chang, "Expert Systems in Industrial Engineering," International Journal of Production Research, 1986 Vol 24 no 5.
82. Lee, L., "Data Feedback in an Integrated Design and Manufacturing System," PhD Thesis, Dept of Manufacturing Engineering, Loughborough University of Technology, 1990.
83. Li, R.K. and S. Adiga, "Part Feature Recognition System - a Vital Link in the Integration of CAD and CAM," Proc. 9th International Conference on Production Research, 1987.
84. Liang, G.R. and C.R. Liu, "Automatic NC Programming as a Decomposable Frame Problem," Integrated and Intelligent Manufacture, ASME, 1986.



85. Lillehagen, F.M. and T. Dokken, "Towards a Methodology for Constructing Product Modelling Databases in CAD," File Structures and Data Bases for CAD, IFIP, North Holland, 1982.
86. Link, C.H., "CAPP-CAM-I Automated Process Planning System," Proceedings of the CAM-I NC Conference, 1976.
87. Lu, S., J.B. Thompson, and M. Klein, "A Cooperative Product Development Environment to Realise the Simultaneous Engineering Concept," Proceedings of Computers in Engineering Conference, ASME, 1989.
88. Magill, W.R. and A.J. McLeod, "Automated Generation of NC Part Programs from a Feature Based Component Description," International Journal of Computer Integrated Manufacturing, Vol 2, No.4 1989.
89. Mantyla, M., J. Opas, and J. Puhakka, "A prototype System for Generative Process Planning of Prismatic Parts," Proc. IFIP Conference on Advances in Production Management Systems, North Holland, 1987.
90. Mantyla, M., An Introduction to Solid Modelling, Computer Science Press, 1988.
91. Mantyla, M., "Directions for Research in Product Modelling," Computer Applications in Production and Engineering, North Holland, 1989.
92. Matsushima, K., N. Okada, and T Sata, "The Integration of CAD and CAM by Application of Artificial Intelligence Techniques," Annals of CIRP, 1982.
93. McKay, A., "User23 - The Tooling Database Maintenance Routines," GMP Project, Dept. of Mechanical Engineering, University of Leeds, 1987.
94. McKay, A., "A Framework for the Project Meta-Structure," ISS-PDS-Report 8, Dept of Mechanical Engineering, Leeds University , 1989.
95. McKay, A., P. Bell, and R. Young, "The Description of Planned Processes," ISS Project, Dept. of Mechanical Engineering, University of Leeds; Dept. of Manufacturing Engineering, Loughborough University of Technology., 1990.
96. Melkote, R. and D.L. Taylor, "An Implementation of Rule Based Selection of Milling Cutters, Feed Rates and Spindle Speed," ASME International Computers in Engineering Conference, 1988.

97. Menassa, R.J. and W.R. DeVries, "Locating Point Synthesis in Fixture Design," Annals of CIRP, 1989.
98. Metcut,, "Cutdata," Metcut Research Associates, Cincinnati, Ohio, 1985.
99. Metcut, Research Associates Inc., Machining Data Handbook, Metcut Research Associates Inc., 1986.
100. Midha,, "Macbank," PERA Report 375, 1983.
101. Mill, F., P. Husbands, and S. Warrington, "A Knowledge Based Process Planning System," Proc. 2nd. International Conference on the Application of AI in Engineering, Computational Mechanincs Publications, Boston, 1987.
102. Mullineux, G., CAD: Computational Concepts and Methods, Kogan Page, London, 1986.
103. Nau, D.S. and T.C. Chang, "Prospects for Process Selection using AI," Computers in Industry, North-Holland, 1983.
104. Nevins, J.L. and D.E. Whitney, Concurrent Design of Products and Processes, McGraw Hill, 1989.
105. Norwood, D.L., "ICAM factory of the future concept," CASA/SME Autofact 5 conference, Nov. 1983.
106. Oliver, R.G., "Hierarchical Data Structures for Solid Modelling," PhD Thesis, University of Leeds, 1987.
107. Pafec,, Dogs NC User Manual level3.2, Pafec Ltd, Srelley Hall, Nottingham NG8 6PE, 1987.
108. PDES,, "Product Data Exchange Specification, Firrt Working Draft, NISTIR 88-4004," National Institute of Standards and Technology, National Engineering Laboratory, Gaithersburg, MD 20899, Dec. 1988.
109. PDES/STEP,, "Manufacturing - Process Plan (working paper) Document 4.1.3.2.1, version 3.2," Manufacturing Technology Committee, Jan. 1991.
110. PERA,, "Machining Data Bank," PERA Report 324, 1977.
111. PERA,, "Machining Data Bank," PERA Report 336, 1978.
112. PERA,, "Computerised Machining Data Bases," PERA Report 393, 1985.



113. Pinte, J., "TNO/CRIF Proposal for Follow-up Study of the KEE Implementation of the XPS-2 Knowledge Base Study," CAM-I Meeting, Ref M-90-PP-03, 1990.
114. Pratt, M.J., "Solid Modeling and the Interface between Design and Manufacture," Computer Graphics and Applications, IEEE, 1984.
115. Pratt, M.J., "Synthesis of an Optimal Approach to Form Feature Modelling," Proceeding of Computers in Engineering Conference, ASME, 1988.
116. Preiss, K. and E. Kaplansky, "Automated Part Programming for CNC Milling by Artificial Intelligence Techniques," Journal of Manufacturing Systems, vol. 4, 1985.
117. Ranyak, P.S. and R. Fridshal, "Features for Tolerancing a Solid Model," Integrated and Intelligent Manufacturing, ASME, 1988.
118. Rapello, R.G., Essentials of Numerical Control, Prentice/Hall , 1986.
119. Report, C-85-PPP-03., "Part Features for Process Planning," CAM-I International, Nov. 1986.
120. Requicha, A.A.G. and H.B. Voelcker, "Solid Modeling: A Historical Summary and Contemporary Assessment," Computer Graphics and Applications, IEEE, 1982.
121. Requicha, A.A.G. and H.B. Voelcker, "Solid Modeling: Current Status and Research Directions," Computer Graphics and Applications, IEEE, 1983.
122. Rich, E., "Artificial Intelligence," McGraw Hill, 1983.
123. Ross, D.T., "Structured Analysis (SA): a Language for Communicating Ideas," IEEE Transactions on Software Engineering, Jan 1977.
124. Ross, D.T., "Structured Analysis for Requirements Definition," IEEE Transactions on Software Engineering, Jan 1977.
125. Roy, U. and C.R. Liu, "An Expert System Approach to the Machine Tool Sequencing and Machining Sequencing Problems in Computer Aided Process Planning," School of Industrial Engineering, Purdue University, 1988.
126. Sata, T., S. Kawabe, S. Shimura, K. Miyashita, and F. Kimura, "Programming for Machining Based on Work-piece Models in Computer," Annals of CIRP, 1983.



127. Sata, T., F. Kimura, M. Inui, and H. Shuzuki, "Generation and Verification of Process Plans using Dedicated Models of Products in Computers," Knowledge Based Expert Systems in Manufacturing, ASME, 1986.
128. Schaffer, G., "GT via Automated Process Planning," American Machinist, May 1980.
129. Shah, J.J. and D. Miller, "A Structure for Integrating Geometric Tolerances with Form Features and Geometric Models," Proceedings of Computers in Engineering Conference, ASME, 1989.
130. Shaw, N.K., M.S. Bloor, and A. De Pennington, "Product Data Models," Research in Engineering Design, 1989, Vol.1 No.1.
131. Shigley, J.E., Mechanical Engineering Design, McGraw-Hill, 1986.
132. Smith, G.W. and M. Wang, "Modelling CIM systems Part 2: the generic functions," Computer Integrated Manufacturing Systems, August 1988.
133. Spur, G., F.L. Krause, and P Armbrust, "Product Models as a Basis for Integrated Design and Manufacturing," International Journal of Machine Tool Design and Research, Vol 26, No.2 1986.
134. Spur, G., F.L. Krause, and C.M. Lehmann, "Integration of Methods for Knowledge Processing and Geometric Modelling," Effective CAD/CAM, Proceedings of the Institution of Mechanical Engineers, 1989.
135. Spur, G., K. Mertins, and W. Sussenguth, "Integrated Information Modelling for CIM," Proceedings of CIM-CON , 1990.
136. Ssemakula, M.E. and P.O. Sivac, "Automatic Generation of NC Part Programs," Proceedings of Autofact, 1987.
137. STRATA,, Engineering Computer Services Ltd. Litchfield, WS13 6AA., 1990.
138. Tattersall, A., "Planning for Integration - a Data Oriented Approach," Computer Integrated Manufacturing Systems, Vol 1, No.3 1988.
139. Tipnis, V.A., "Research Opportunities in Mechanical Tolerancing," International Workshop, Sponsored by the National Science Foundation, ASME, 1989.

140. van Houten, F.J.A.M, A.H. van't Erve, and H.J.J. Kals, "Part, A Feature Based Computer Aided Process Planning System," Laboratorium voor Productietechniek, Universiteit Twente, 1989.
141. Varvakis, G., PhD Thesis in preparation, Dept of Manufacturing Engineering, Loughborough University of Technology, 1990.
142. Vasilash, G.S., "Simultaneous Engineering," Production, July, 1987.
143. Voelcker, H.B, "Modeling in the Design Process," , 1988.
144. Walker, H. and R. West, "Integrating CAD and CAM through the Feature Based Methodology for Design and Manufacture," Proceedings Matador Conference, 1990.
145. Wang, H.P. and R.A. Wysk, "AIMSI: A Prelude to a New Generation of Integrated CAD/CAM System ," Dept. of Industrial and Management Systems Engineering, Pennsylvania State University, 1986.
146. Weill, R., G. Spur, and W. Eversheim, "Survey of Computer-Aided Process Planning Systems," CIRP Annals, 1982.
147. Wickens, L., "Dimensioning and Tolerancing Engineering Components using Constructive Solid Geometry," SIAM/RPI Conference on Applied Geometry, 1987.
148. Wickens, L.P., "A Syntax for Dimensions and Tolerances," PhD Thesis, Dept of Mechanical Engineering, Leeds University, 1990.
149. Willis, D., I.A. Donaldson, A.D. Ramage, J.L. Murray, and M.H. Williams, "A Knowledge Based System for Process Planning based on a Solid Modeller," Heriot-Watt Universtiy, 1988.
150. Woo, T.C., "Computer Aided Recognition of Volumetric Designs," Advances in Computer-Aided Manufacture, North Holland, 1977.
151. Woodwark, J.R., "Some Speculations on Feature Recognition," IBM UKSC Report 186, 1988.
152. Wysk, R.A., M.Y. Barash , and C.M. Moodie, "Unit Machining Operations: an Automated Process Planning and Selection Programme," Trans. of ASME, Journal of Eng. for Industry., vol. 102, Nov. 1980.

153. Wysk, R.A., "Automated Process Planning Systems - An Overview of Ten Years of Activities," 1st CIRP working seminar on CAPP, Jan. 1985.
154. Yoshimura, M., K. Itani, and K. Hitomi, "Integrated Optimization of Machine Product Design and Process Design," International Journal of Production Research, Vol 27, No8, 1989.
155. Young, R., "User17 - The Structure Editor Menu Option - Tool Selection," GMP Project, Dept. of Mechanical Engineering, University of Leeds, 1987.
156. Young, R.I.M. and R. Bell, "Fixturing Strategies and Geometric Queries in Setup Planning," International Journal of Production Research, vol. 29, no 3, 1991.
157. Yuen, M.F., S.T. Tan, W.S. Sze, and W.Y. Wong, "An Octree Approach to Rough Machining," Proceedings Institute of Mechanical Engineers, Vol 201, 1987.
158. ISS, Group, "ISS-user-5, User Manual for demonstration software," Department of Mechanical Engineering, Leeds University and the Department of Manufacturing Engineering, Loughborough University of Technology, 1990.



## APPENDIX 1.

### INFORMATION SUPPORT SYSTEMS FOR DESIGN AND MANUFACTURE (SOFTWARE UTILITIES AND ACTIVITY MODELLING)

#### 1.1 INTRODUCTION

Information Support Systems for Design and Manufacture was a research programme pursued at Loughborough University of Technology and Leeds University. It was an ACME and DTI funded project, supported by a group of industrial collaborators. This appendix describes the software utilities used in the research as well as the IDEF0 Modelling technique used by the research team.

#### 1.2 PRODUCT DATA REPRESENTATION USING THE STRUCTURE EDITOR

##### 1.2.1 Building Data Models.

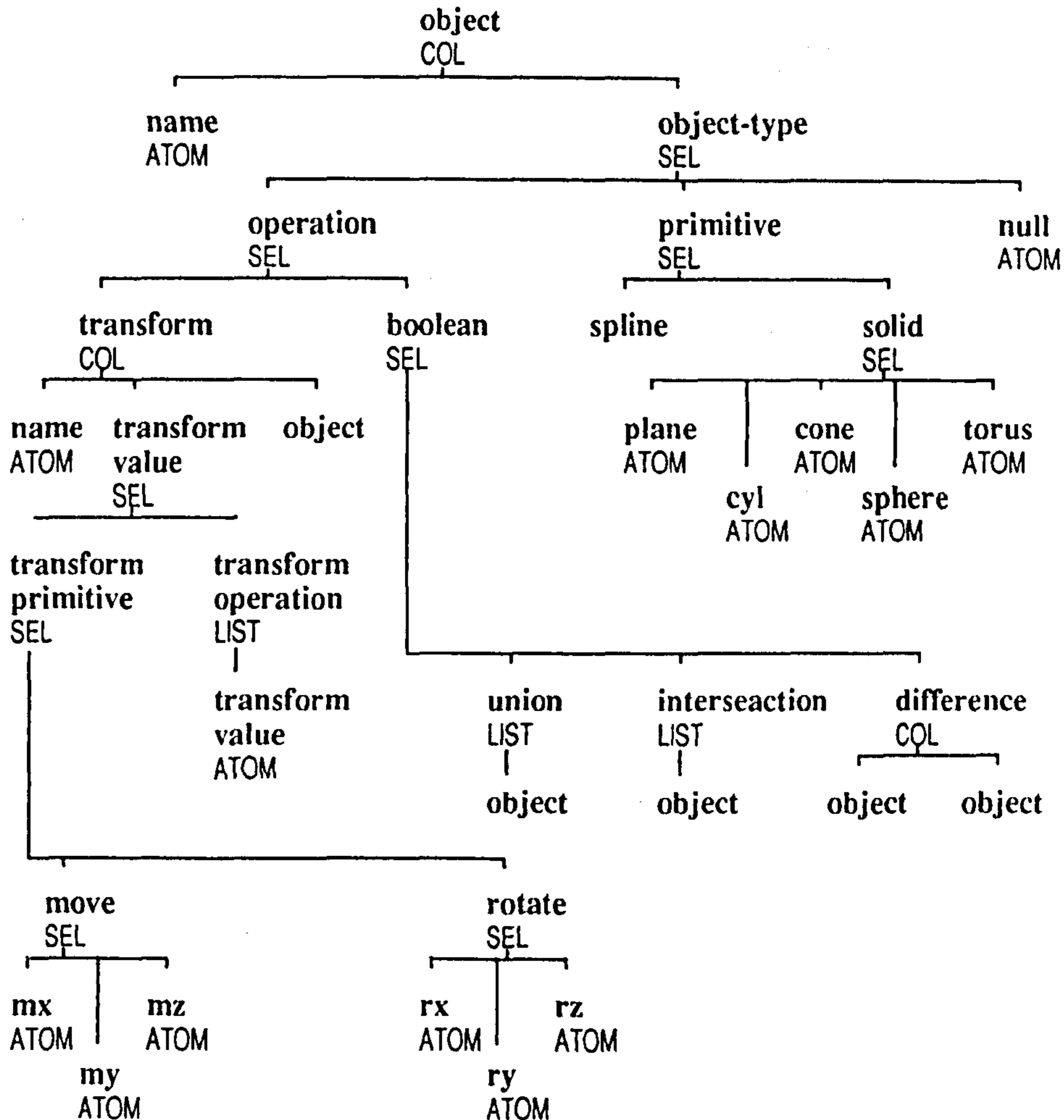
The Structure Editor(SE) [16] provides an ability to build and edit data structures. These Structures can be created from a basic set of building blocks which are called collections, selections, lists and atoms. These represent typical software language constructs such as records, enumeration types, linked lists, integers, real numbers and strings as described in the ADA language reference manual [66].

"Collections" are equivalent to ADA records and describe a group of one or more unlike things. "Lists" are equivalent to ADA access types and are comprised of zero, one or many like things. "Selections" are the equivalent of ADA enumeration types and describe a choice from different things. The choice is made from a restricted range of pre-determined alternatives which comprise the set of possible selections. "Atoms" are the equivalent to some of the pre-defined types in ADA and describe the lowest level of structure, i.e. the form of the leaf nodes in the structure. These can be "integer" numbers, "real" numbers, "names", "comments" or "nil" atoms. "Names" and "comments" are both character strings, but the former is a string of pre-determined length which has some meaning when used while the latter has no pre-determined length and is intended to be used for general descriptive purposes only. "Nil" atoms leave the node with an empty value.

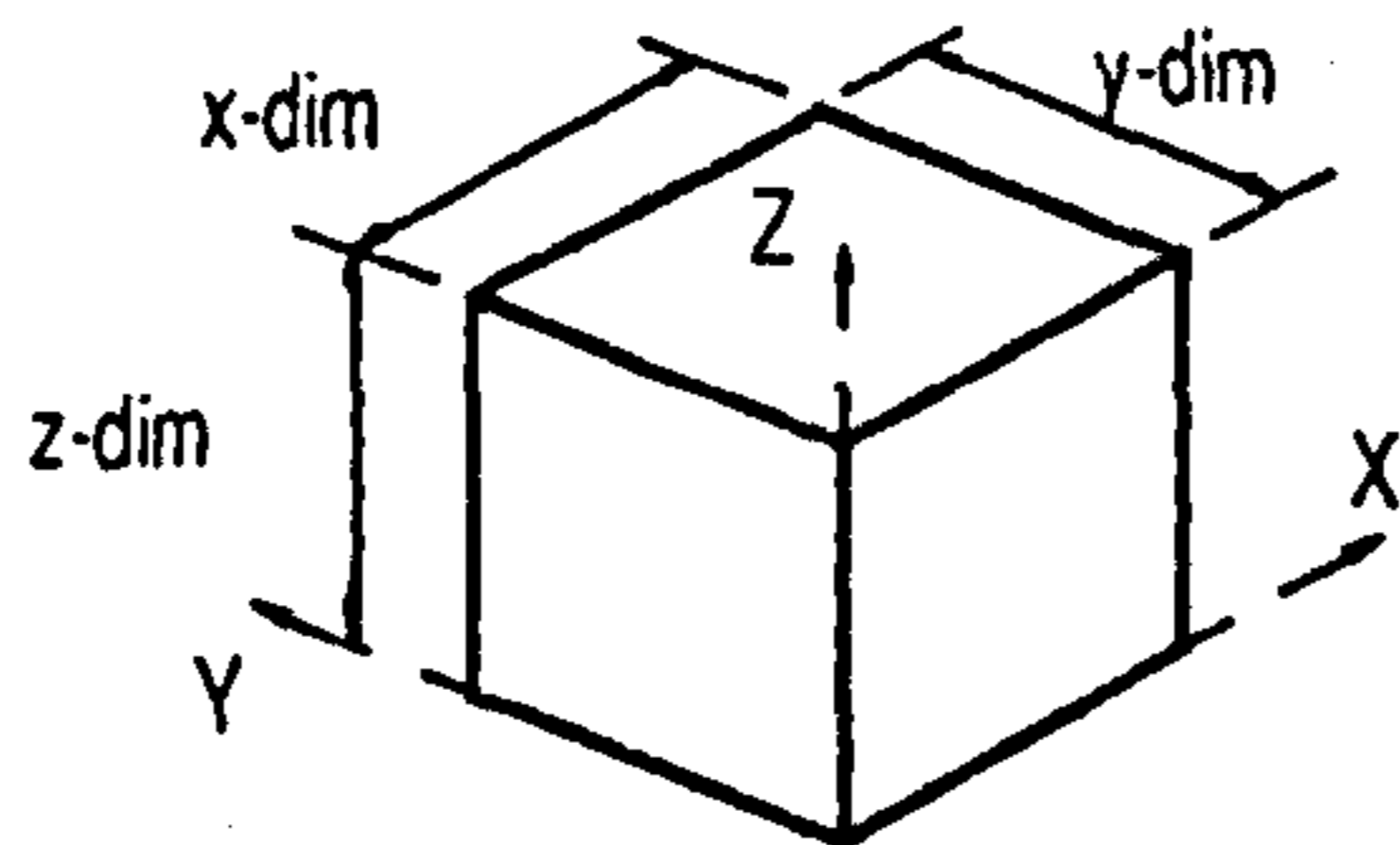
These constructs provide the means by which the structure of a data model can be described, such as a Product Data Model. Figure A1.1 provides an example of the use of such constructs to produce a structure to represent a solid object.

##### 1.2.2 Building Models from Data Models.

Once the structure of the data has been defined as a data model, as described in the previous section, models of specific instances of the data can be produced. This is best explained by the use of an example. The data model of an object, as illustrated in figure A1.1 can be used to represent specific instances of objects. For example typical Constructive Solid Geometry primitives such as block, sphere and cone can be described. Figure A1.2 shows how the object data model can be used to describe a block. Models of products can be described by following the same principles.



**An Example Object Data Model using SE Building Blocks  
Figure A1.1**



Block: intersection  
 +xface : ry = 90, mx = x-dim : plane  
 -xface : ry = -90 : plane  
 +yface : rx = -90, my = y-dim : plane  
 -yface : rx = 90 : plane  
 +zface : mz = z-dim : plane  
 -zface : rx = 180 : plane

**An Example Instance from the  
Object Data Model  
Figure A1.2**



### 1.2.3 Interfacing to Software Applications.

The use of data from a Product Model in software applications requires interfaces to be written to extract data from the model for use by the software application program. Also, data generated by an application, which has implications for the product, should be written back to the Product Model.

These requirements are aided by an interface generation capability within the Structure Editor which produces ADA software source code which can be edited by the system builder to produce interfaces. This ADA code forms the basis for traversing the Product Model and extracting data which can then be converted into the form needed by the Application.

### 1.2.4 Parametric Representations.

The SE provides a capability which allows the representation of pieces of structure in terms of parameter values. For example the 3 parameters of length, width and depth could be specified. Volume could then be represented as being dependent on these 3 through the relationship

$$\text{volume} = \text{length} * \text{width} * \text{depth}$$

by changing the parameter values, different values for dependent variables can be evaluated. e.g:

```
PARAMETERVALUEVALUE
length 4 3
width 3 2
depth 2 1
DEP.VARIABLE:
volume 24 12
```

This facility has proved particularly valuable in exploring the association of manufacturing data with geometric shapes to define manufacturing features.

### 1.2.5 Sharing Data

A further capability of the SE is the ability to share data in different pieces of structure such that if one value is changed the other value will automatically change also. For example if a component is defined as containing 2 holes of equal diameter, then the diameter information for the 2 holes should be shared. If the diameter of one of the holes is then changed, the diameter of the other hole will change automatically.

## 1.3 DATA MODEL UTILITIES USED IN THE RESEARCH.

### 1.3.1 General Data Structures.

This section describes those data structures and related software utility functions which were available with the SE, for use in the research described in this thesis. These are related to the representation of numeric data, geometry and feature geometry

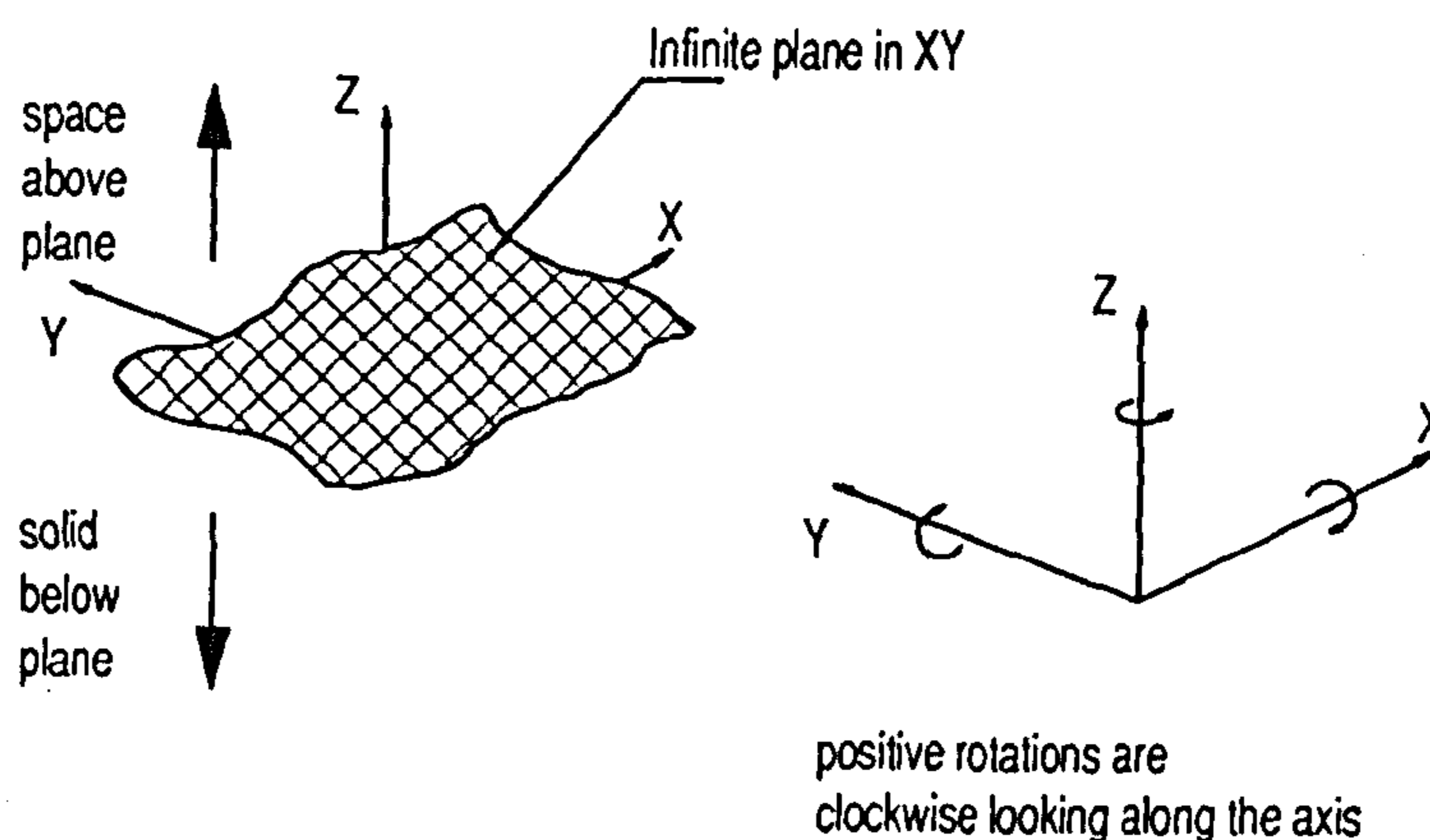


### 1.3.2 Numeric Representation and Evaluation.

In modelling products there is a need to represent arithmetic and mathematical expressions. A data model structure to represent such Numeric Expressions as addition, subtraction, multiplication and division, as well as simple trigonometric functions was available. The calculated value resulting from any particular numeric expression could be found using a numeric evaluation software application. Tolerance values were also defined within the Numeric Expression structure. A tolerance value is represented as a nominal value with an upper tolerance and a lower tolerance. This provided a simple basis for the representation of dimensional tolerance data.

### 1.3.3 Geometric Representation and Evaluation.

Solid Model representations of the component under consideration and of feature shapes are used in this research. These representations have been built using the geometric data



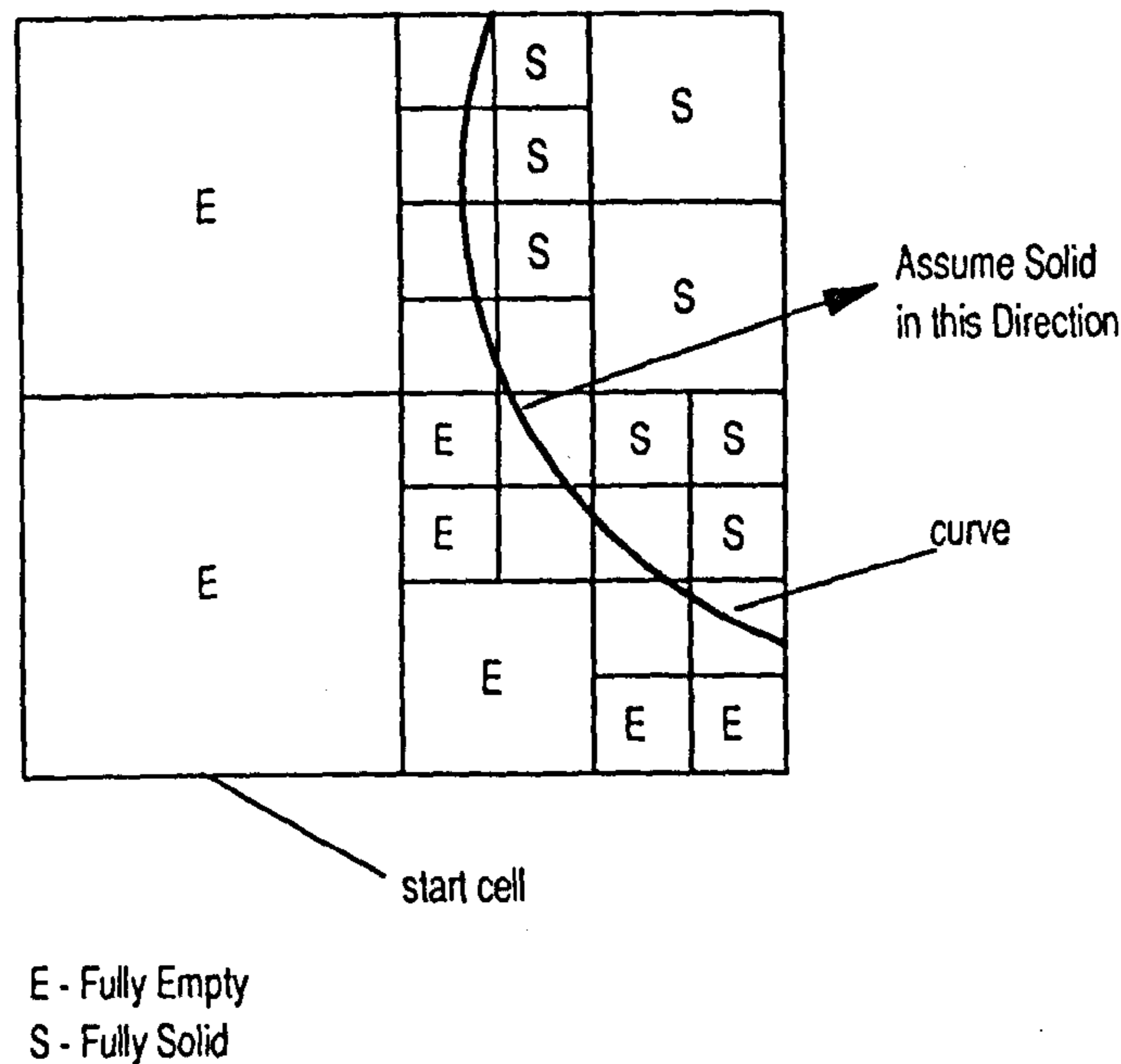
**Planar half-space and rotation definitions**  
**Figure A1.3**

model illustrated in figure A1.1 as a basic geometric structure. This enables typical Constructive Solid Geometry operators and transforms to be used in building a model. It also enables shapes to be represented using half space definitions. An example half-space definition for a plane, and the definition for rotations, are illustrated in figure A1.3. Similarly, definitions exist for other half-space types.

The geometric data model allows geometric shapes to be described and positioned in space. The evaluation of the geometric representation which can provide data describing how a piece of geometry relates to its surrounding environment is provided using a Spatially Divided Solid Model (SDSM). This provides a cellular decomposition of a geometric representation [105], where each cell can provide data on whether it is inside a solid, outside a solid, or on the surface of a solid. Also it provides its position in space and points to its neighbouring cells. Where a cell lies on a surface, pointers are held which point to the faces contained in the cell.

The SDSM cell decomposition starts from a cube which totally encloses the solid under consideration. The cube is then sub-divided into 8 equal cubes. Each of these is checked to identify whether it contains a surface. If so, it is further decomposed. This cycle is repeated to a level defined by the user, or by the software application using the SDSM. Decomposition is usually performed to between 5 and 7 levels. Level 5 provides a reasonable decomposition fairly quickly, while level 7 produces the SDSM to a smaller cell size and therefore to much greater detail. Figure A1.4 shows a 2 dimensional representation of an SDSM

cell decomposition showing how cells containing faces are further decomposed, while cells



## A 2-dimensional cell decomposition

Figure A1.4

which are fully solid or fully empty are not.

### 1.3.4 Feature Representation and Evaluation.

The Structure Editor allows features to be defined using its parametric capability. This enables features to be built in terms of the parameters which influence them, hence providing generalised feature shape and manufacturing definitions rather than definitions with specific dimensions.

The Structure Editor allows features to be described as named, transformed or aggregate features. The named feature allows a descriptive name to be associated with a feature, while a transformed feature allows a geometric transform to be applied to a feature to change its position in space. Aggregate features allow a new feature to be built from already available feature definitions. For example a group of holes can be defined as an aggregate of a number of single holes. The single hole definition is used and transformed to each of the relevant positions in the new aggregate feature.

Features can be added to a geometric description to introduce them onto the component being described. The Structure Editor enables this through a solid description capability called a Solid Application. This provides a data structure which is defined as a piece of solid geometry to which can be added any number of features. As features are added to a component description the values to be associated with each parameter is input by the user. Once this has been done a software utility can be used, which takes these supplied parameter values and evaluates each field in the data structure which involves their use. In this way the actual definition of a specific component description can be found.

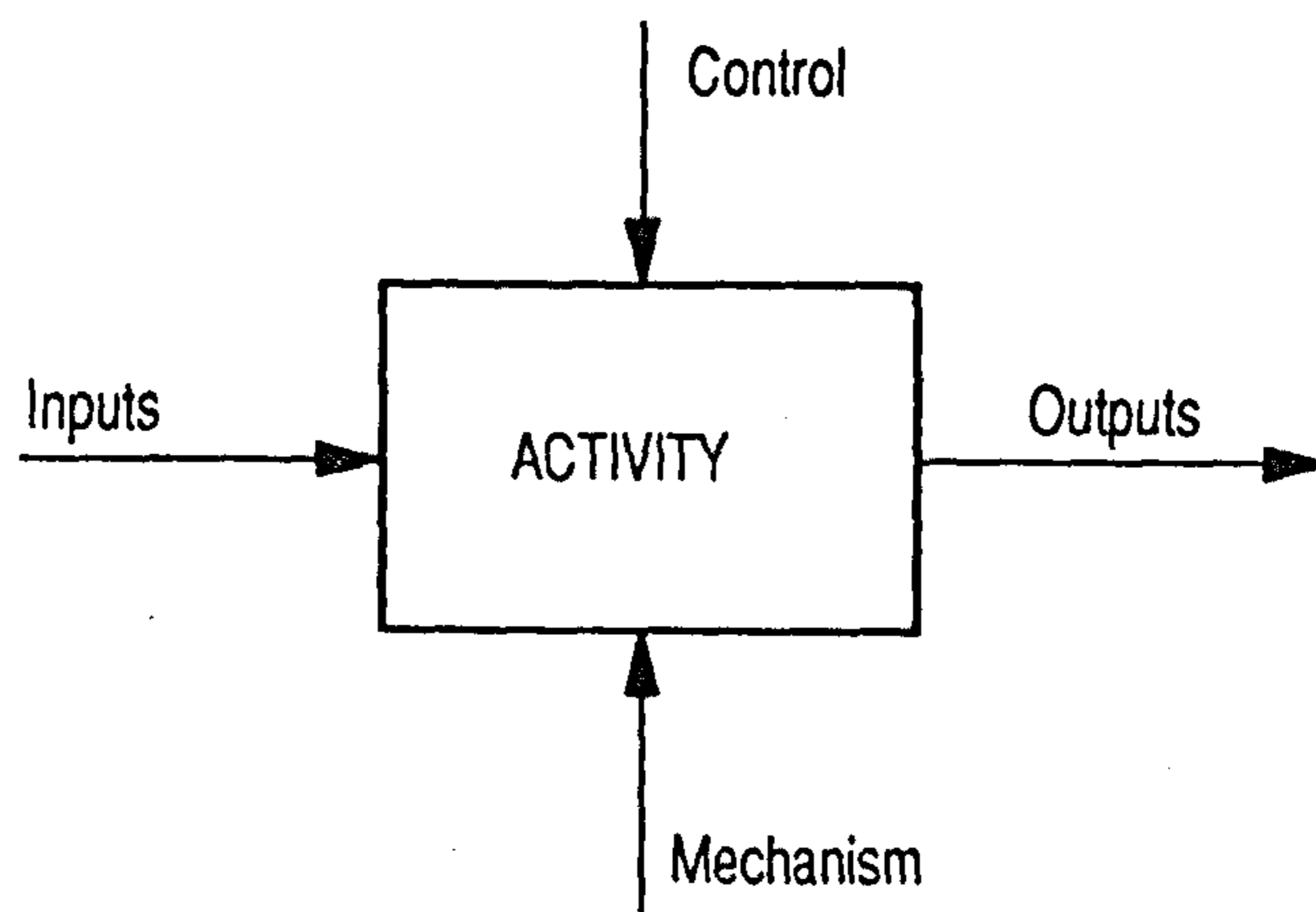


#### 1.4 IDEF0 - A FORMAL METHOD OF EXPLORING ACTIVITY RELATIONSHIPS

Activity modelling provides a way of representing activities, the data which they require, the data which they produce, and their relationships to other activities. By performing this type of modelling there is a greater chance of success in providing a valid representation of activity relationships and their data flows.

One method of representing activities and interrelationships is IDEF0, which stands for ICAM Definition language, a representation technique based a Structured Analysis Design Technique (SADT) [123,124]. The name follows from the US airforce ICAM programme, where the technique was used to consider and overall concept for CIM [105]. The technique has also proved valuable in industry in aiding the understanding of how aspects of companies' businesses operate [80]. IDEF0 has been used successfully in research programmes and industrial applications to model manufacturing activities.

A IDEF0 model consists of boxes and arrows coming in and out of each box as shown in



#### IDEF0 Notation

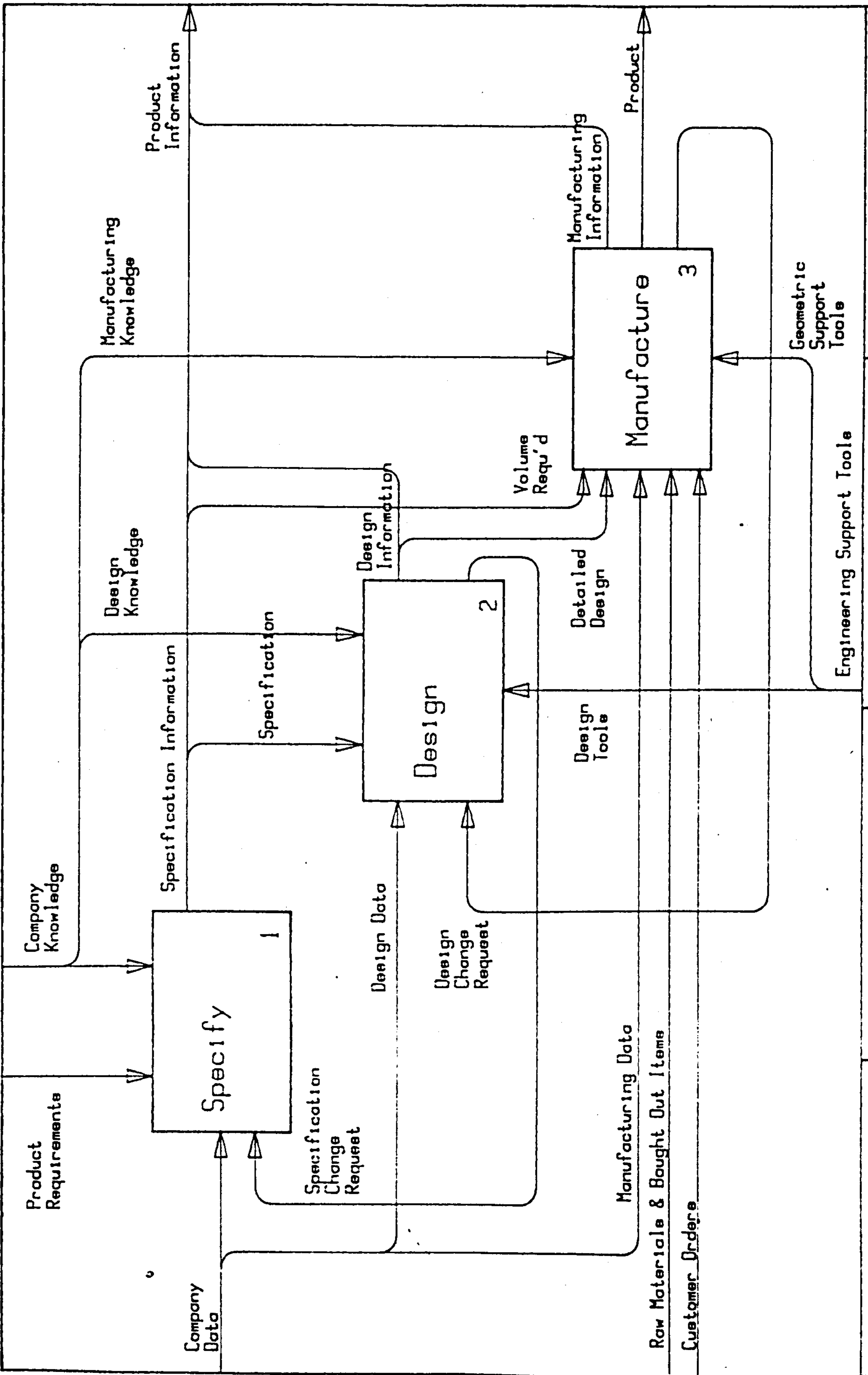
Figure A1.5

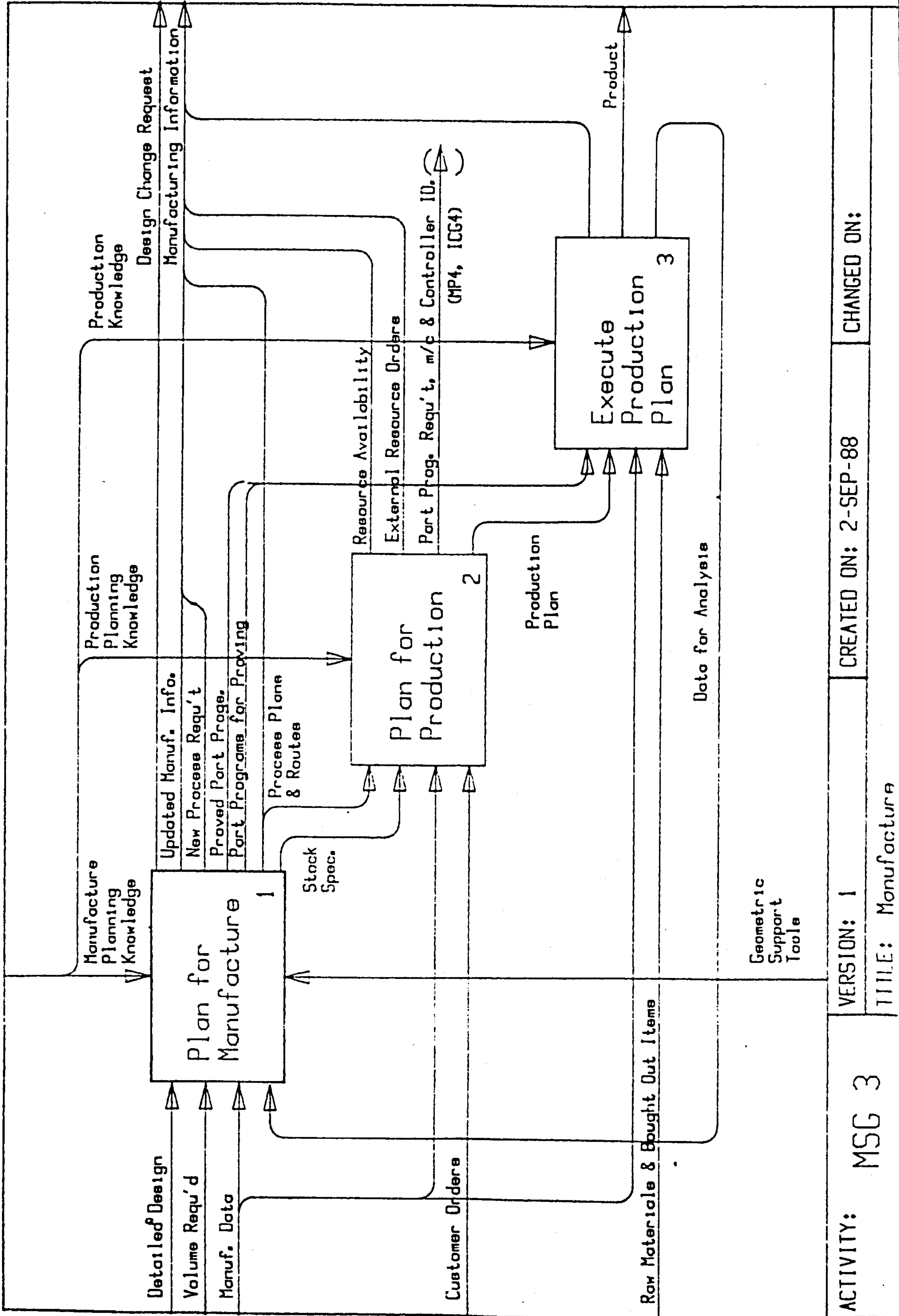
figure A1.4. The box is used to represent the activity to be performed. The arrows from the left describe the inputs to the activity. Arrows into the top also describe inputs, but inputs which have a controlling effect on the activity. Arrows into the bottom identify the means by which the activity is performed and arrows coming out from the right describe the outputs from the activity. An individual activity can be broken into sub-activities and each sub-activity can be further sub-divided. This can be repeated as many times as suits the purposes of the model developers. An important rule in the use of IDEF0 is that all data flows must be accounted for, i.e. any data going into an activity must come from an activity, or be accepted as an input at the top level in the model. Similarly data coming from an activity must go to another. This means that no spurious assumptions can be made concerning data availability and, hence, ensures the consistency of the data requirements.



**APPENDIX 2.**

**AN IDEF0 MODEL OF  
PRODUCING A PRODUCT.**





ACTIVITY: MSG 3

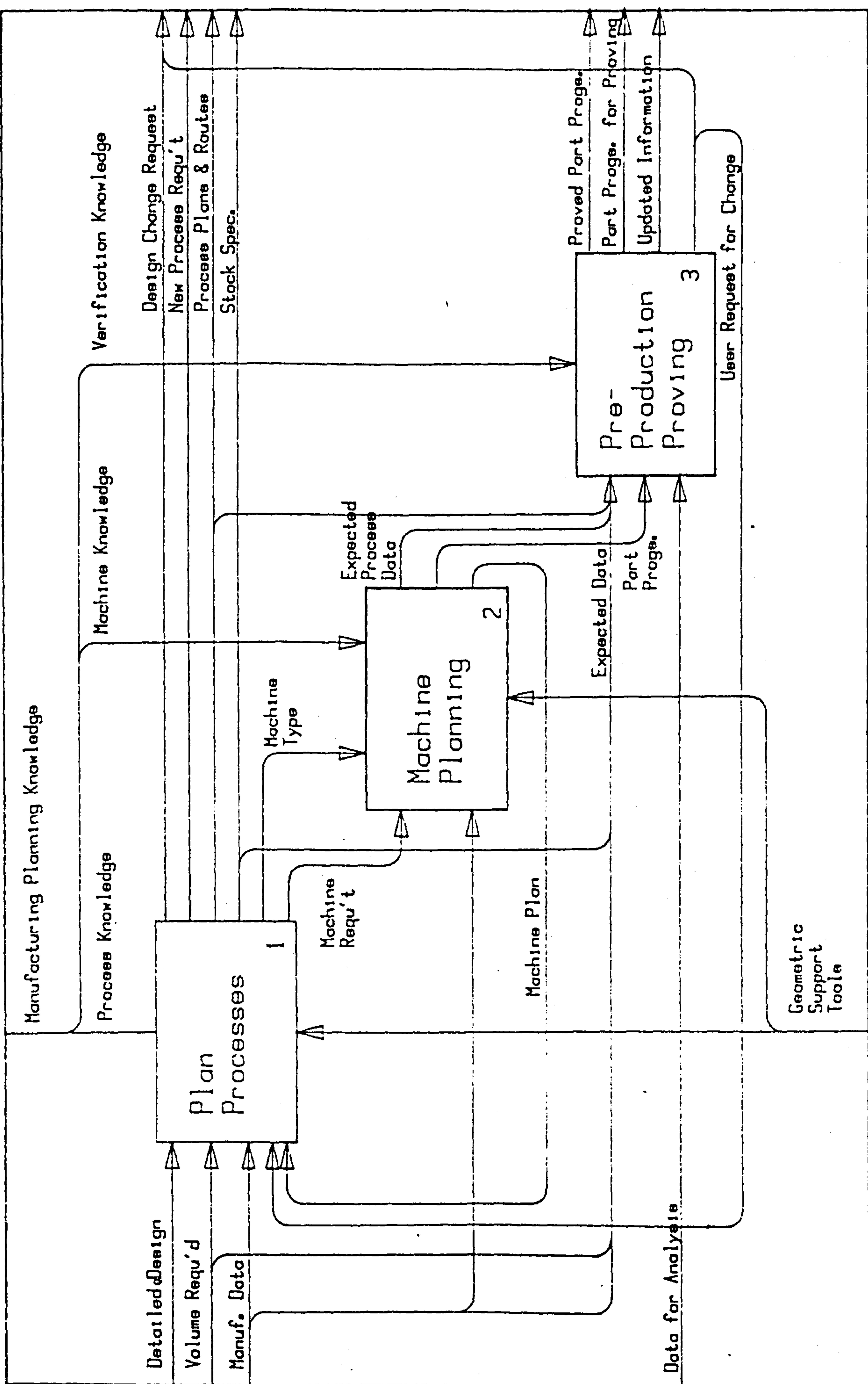
VERSION: 1

TITLE: Manufacture

CREATED ON: 2-SEP-88

CHANGED ON:





Geometric Support Tools

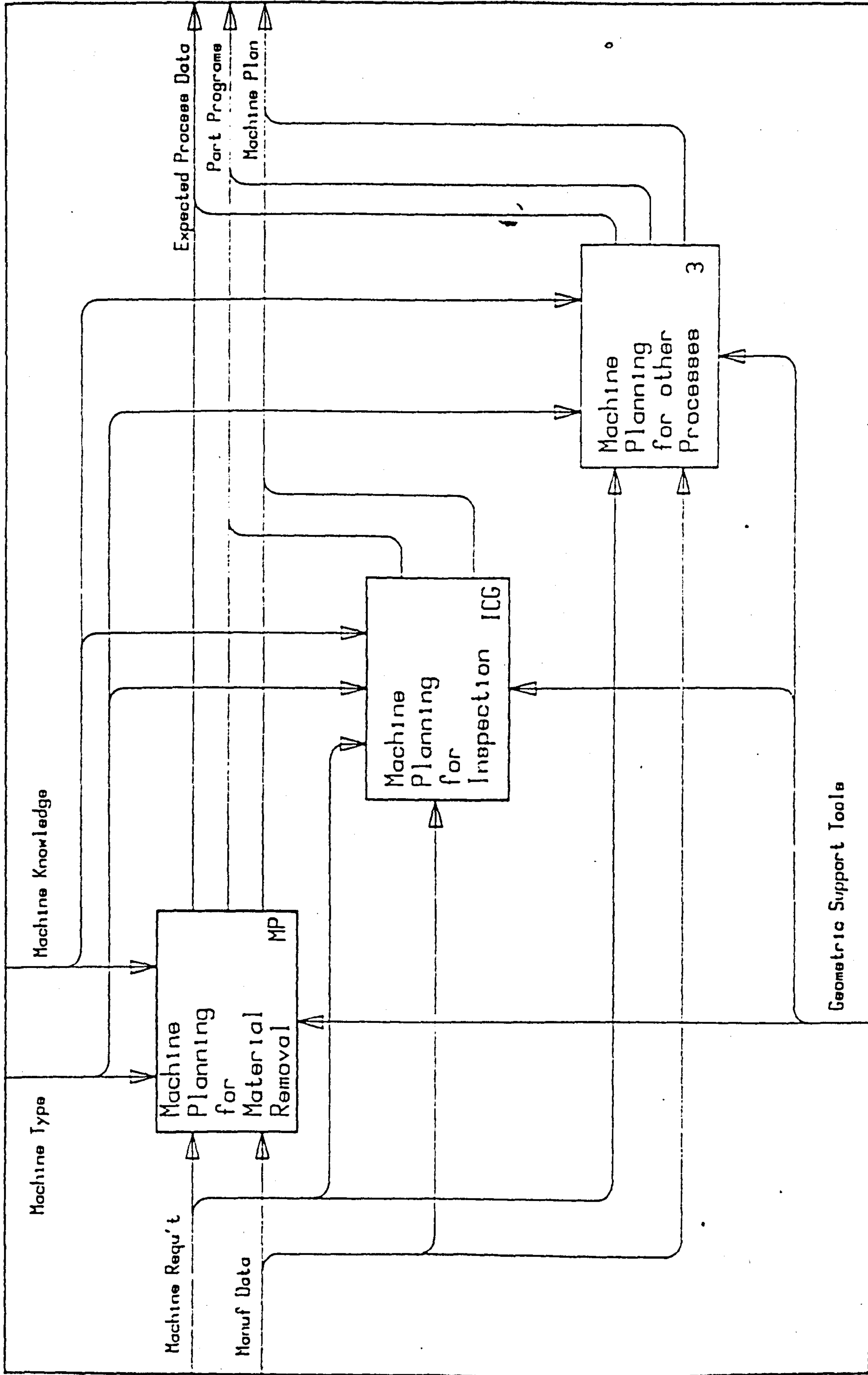
ACTIVITY: MSG31

VERSION: 1

TITLE: Plan for Manufacture

CREATED ON: 2-SEP-88

CHANGED ON:



ACTIVITY:

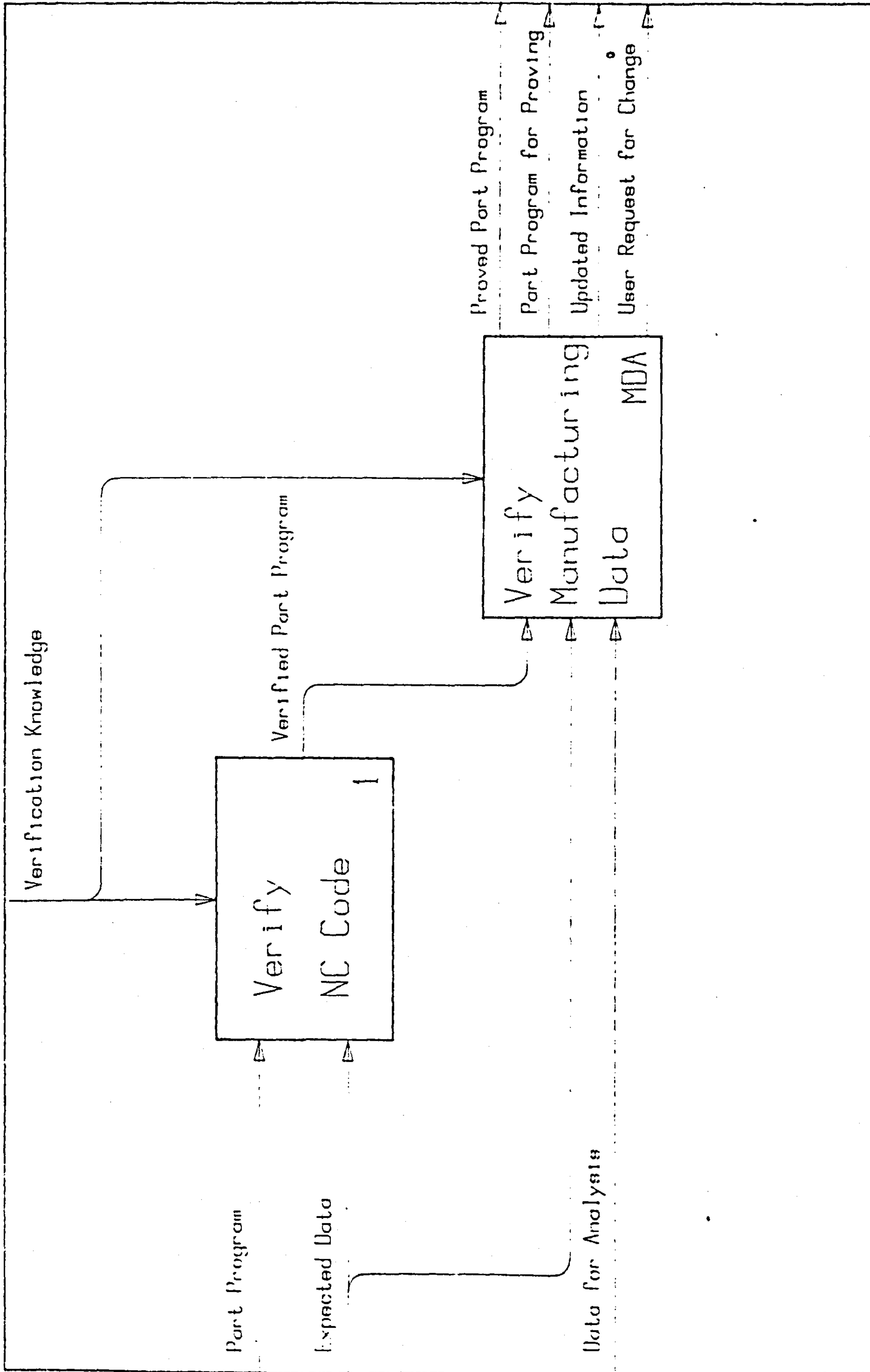
MSG312

VERSION: 1

TITLE: Machine Planning

CREATED ON: 2-SEP-88

CHANGED ON:



ACTIVITY: MSG313	VERSION: 1	CREATED ON: 7-SEP-88	CHANGED ON:
TITLE: Pre-Production Proving			



**PRODUCE PRODUCT - GLOSSARY OF TERMS.**

<b>DESCRIPTION</b>	<b>OUTPUT</b>	<b>INPUT</b>
<b>Company Data:</b> Simple information about the company.		<b>MSG1</b>
<b>Company Knowledge:</b> Information about what the company knows. Often in the form of if 'condition' then 'action'.		<b>MSG1</b>
<b>Customer Orders:</b> Instructions from customers to supply specified goods to a specified deadline.		<b>MSG32</b>
<b>Data for Analysis:</b> A list of data to be analysed.	<b>MSG33</b>	<b>MSG3132</b>
<b>Design Change Request:</b> A request for a specific change in the design to improve/enable manufacture.	<b>MSG311</b>	<b>MSG2</b>
<b>Design Data:</b> Simple information required by designers. e.g. How similar components or products perform.		<b>MSG2</b>
<b>Design Information:</b> All information available to the company concerning design.	<b>MSG2</b>	
<b>Design Knowledge:</b> Information about what the company knows about design. Often in the form of if 'condition' then 'action'.		<b>MSG2</b>
<b>Design Tools:</b> Tools to help designers, e.g. 2D/3D modellers, Features, Graphics, etc.		<b>MSG2</b>
<b>Detailed Design:</b> The description of the components to be produced, in a form suitable for interpretation by Manufacture.		<b>MSG311</b>
<b>Engineering Support Tools:</b> All tools available to support in the Produce Product activity, including Design tools and Geometric Support		<b>MSG</b>

Tools.

**Expected Process Data:**

MSG312

Data generated at the planning stage indicating how the process is expected to proceed. e.g. Tool Life Utilisation, Machine Power required.

**External Resource Orders:**

MSG32

Orders for items required to achieve the product's production.

**Geometric Support Tools:**

MSG311,MSG:  
MSG3122,MSC

Tools to aid in the geometric analysis of the component while planning its manufacture.

**Machine Knowledge:**

MSG312

Information about what the company knows about how to use a particular machine type. Often in the form of if 'condition' then 'action'.

**Machine Plan:**

MSG312

MSG311

Planning information detailing how to do what has been requested on a particular machine.

**Machine Requirement:**

MSG311

MSG312

The description of what is needed of a particular machine. e.g. the pre- and post machined component descriptions, and the batch size.

**Machine Type:**

MSG311

MSG312

The general description of the type of machine being considered.

**Manufacturing Data**

MSG3,MSG31,  
MSG312

Data about the manufacturing environment. e.g. Machine data, tooling, fixturing, machinability, material data.

**Manufacturing Information:**

MSG3

All information available to the company concerning Manufacture.

**Manufacturing Knowledge:**

MSG3

Information about what the company knows about how to manufacture things.

**Manufacturing Planning Knowledge:**

MSG31

Information about what the company knows about how to plan for the manufacture of things.

**New Process Requirement:**

MSG311

The identification that currently available processes (in the company) are not suitable

for the component in question.

**Part Program Requirement,  
Machine and Controller Identification:**

The identification of the specific machine and its controller to be used in the production of a component.

MSG32

MP4,ICG4

**Part Programs:**

Generated NC code which has still to be proven.

MSG312

MSG313

**Part Programs for Proving:**

Part Programs under the control of MDA which are undergoing verification.

MSG3132

MSG33

**Process Knowledge:**

Information about what the company knows about processes.

MSG311

**Process Plans & Routes:**

Approved descriptions of how components should be processed and the machine routes they should follow.

MSG311

MSG32

**Product:**

The item being supplied to a customer.

MSG3

**Product Information:**

All information available about the product.

MSG

**Product Requirements:**

A generalised description of what the product should do.

MSG1

**Production Knowledge:**

Information about what the company knows about production.

MSG33

**Production Plan:**

Details of how and when the company's production capacity is to be used to produce the product.

MSG32

MSG33

**Production Planning Knowledge:**

Information about what the company knows about production planning.

MSG32

**Proved Part Programs:**

Part Programs free from geometrical and technological errors, available for mainstream production.

MSG3132

MSG33

**Raw Materials & Bought Out Items:**

All consumables which need to be sourced from outside the company to meet the production plan.

MSG33

**Resource Availability:**

A description of the utilisation of company equipment.

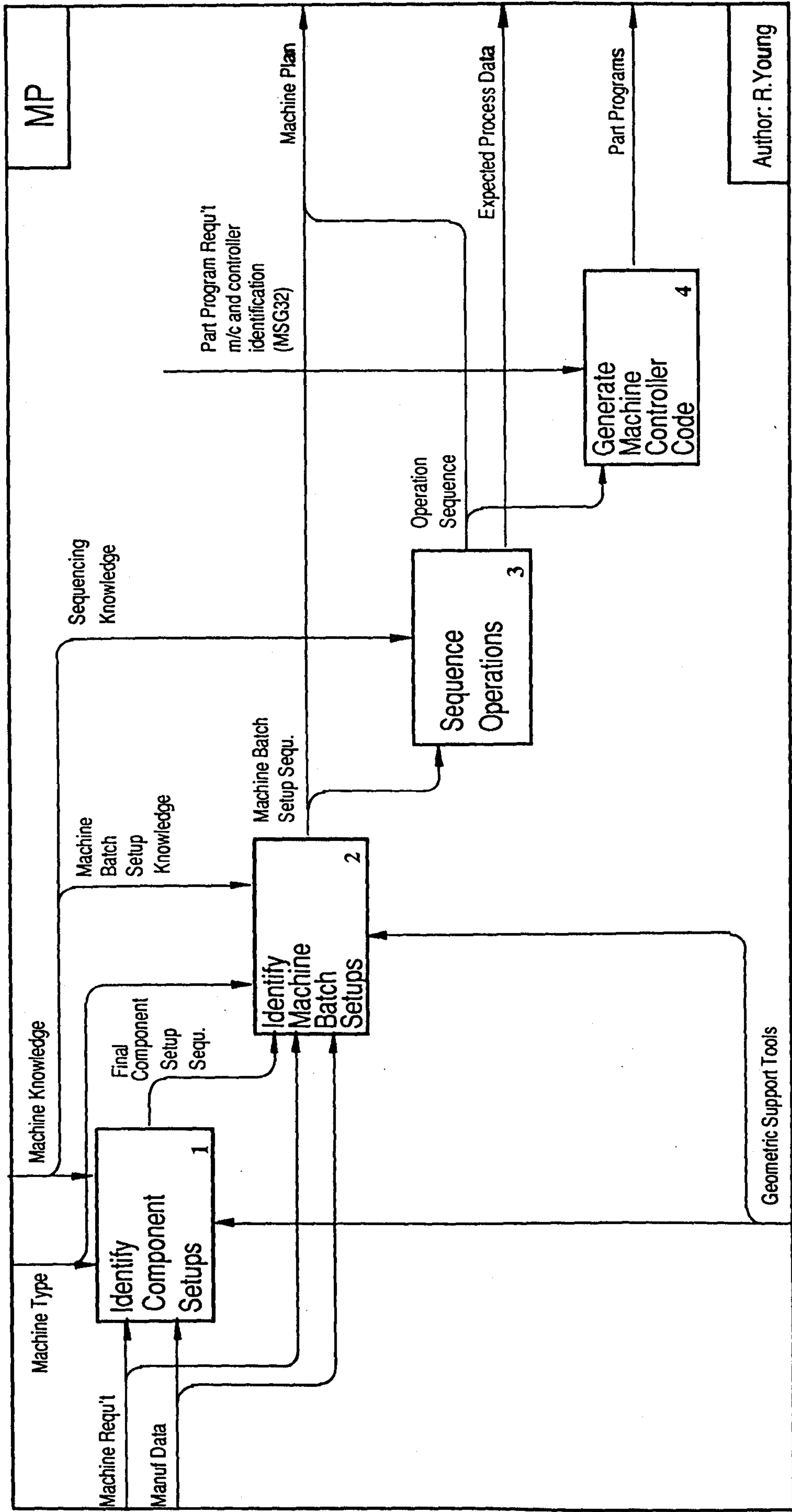
MSG32



<p><b>Specification:</b> A description of the functions the product must perform.</p>	MSG1	MSG2
<p><b>Specification Change Request:</b> A request to change a particular aspect of the specification.</p>	MSG2	MSG1
<p><b>Specification Information:</b> All information generated in the process of producing a specification.</p>	MSG1	
<p><b>Stock Specification:</b> The description of the required raw material.</p>	MSG311	MSG32
<p><b>Updated Manufacturing Information:</b> Manufacturing information changed in the process of Pre-production Proving.</p>	MSG3132	
<p><b>User request for Change:</b> request for changes by the user, in a specific activity.</p>	MSG313	MSG311
<p><b>Verification Knowledge:</b> Information about what the company knows about proving out workpieces.</p>		MSG313
<p><b>Verified Part Program:</b> Part Programs free from geometrical errors.</p>	MSG3131	MSG3132
<p><b>Volume Required</b> The number of items required over a given timescale.</p>		MSG311

**APPENDIX 3.**

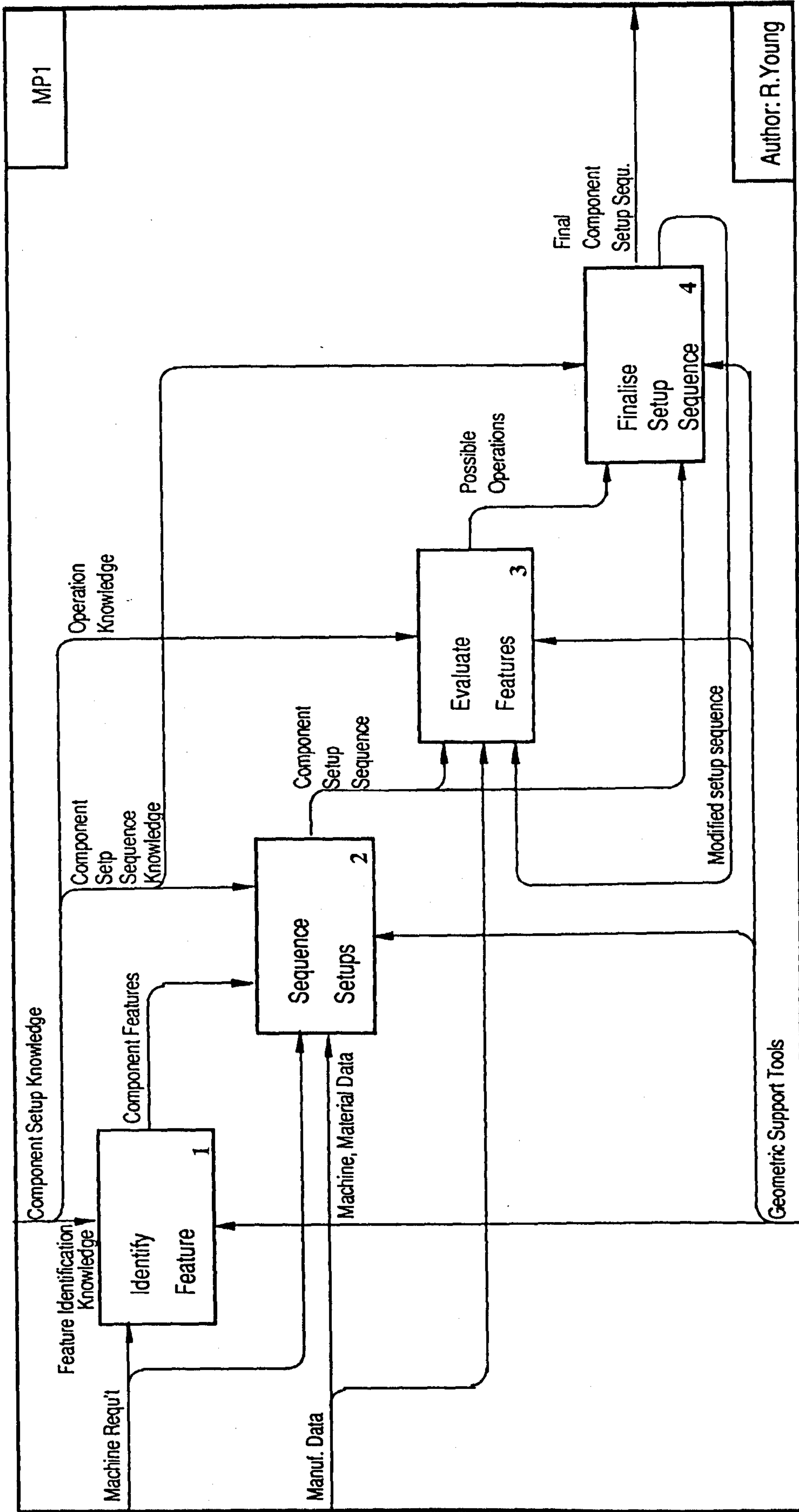
**AN IDEF0 MODEL OF  
MACHINE PLANNING**



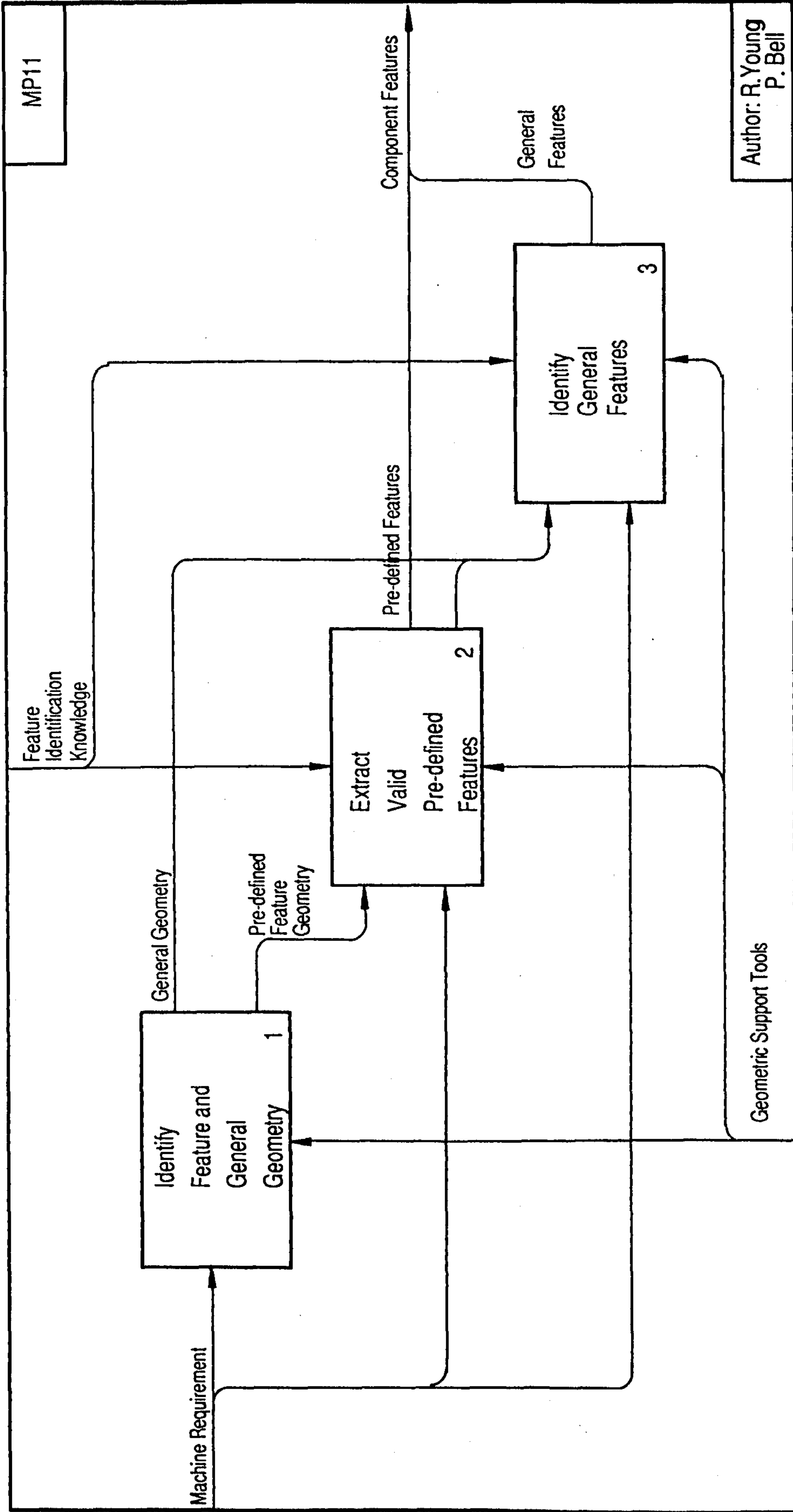
Author: R. Young

### Machine Planning

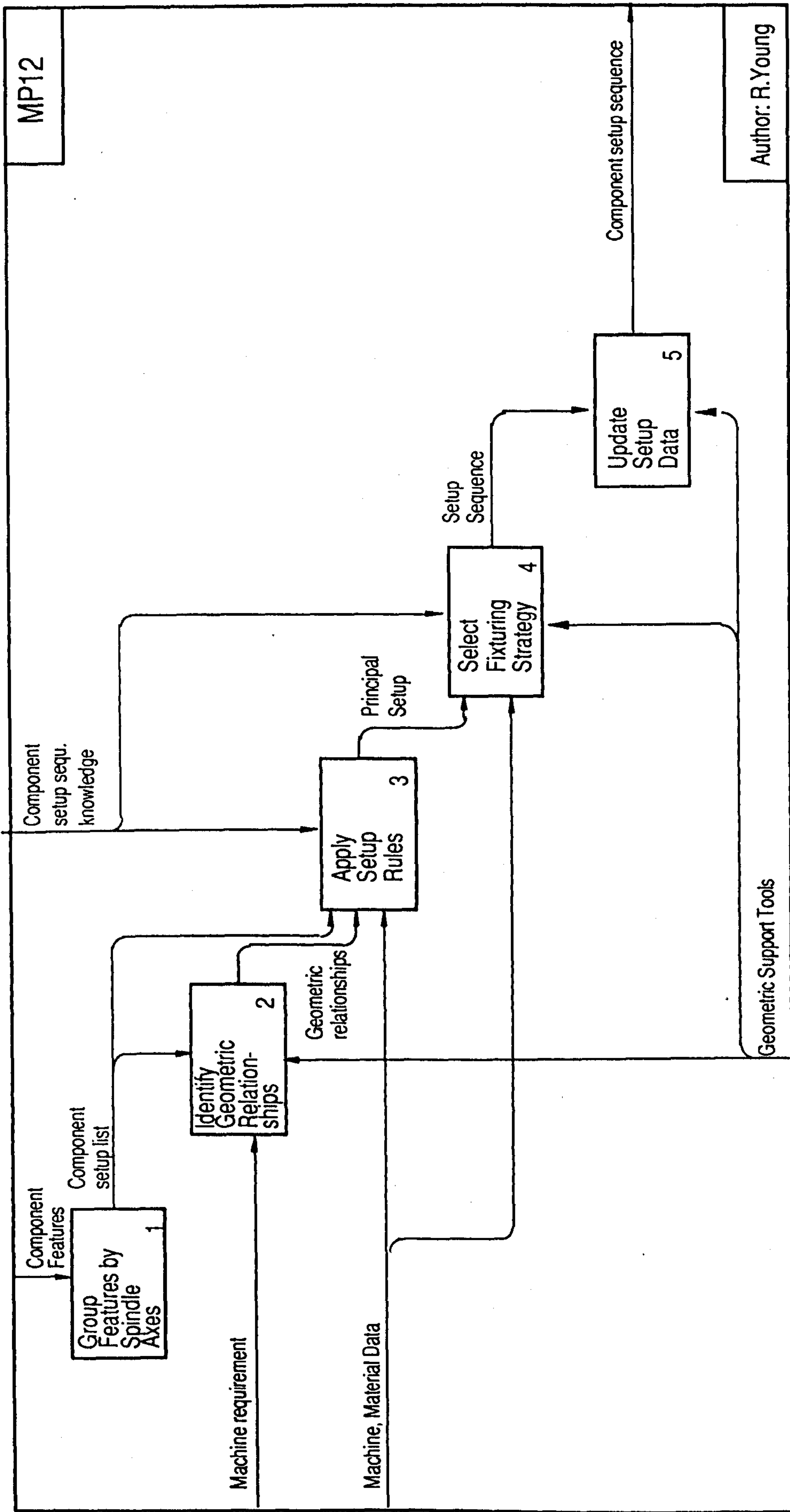




Identify Component Setups



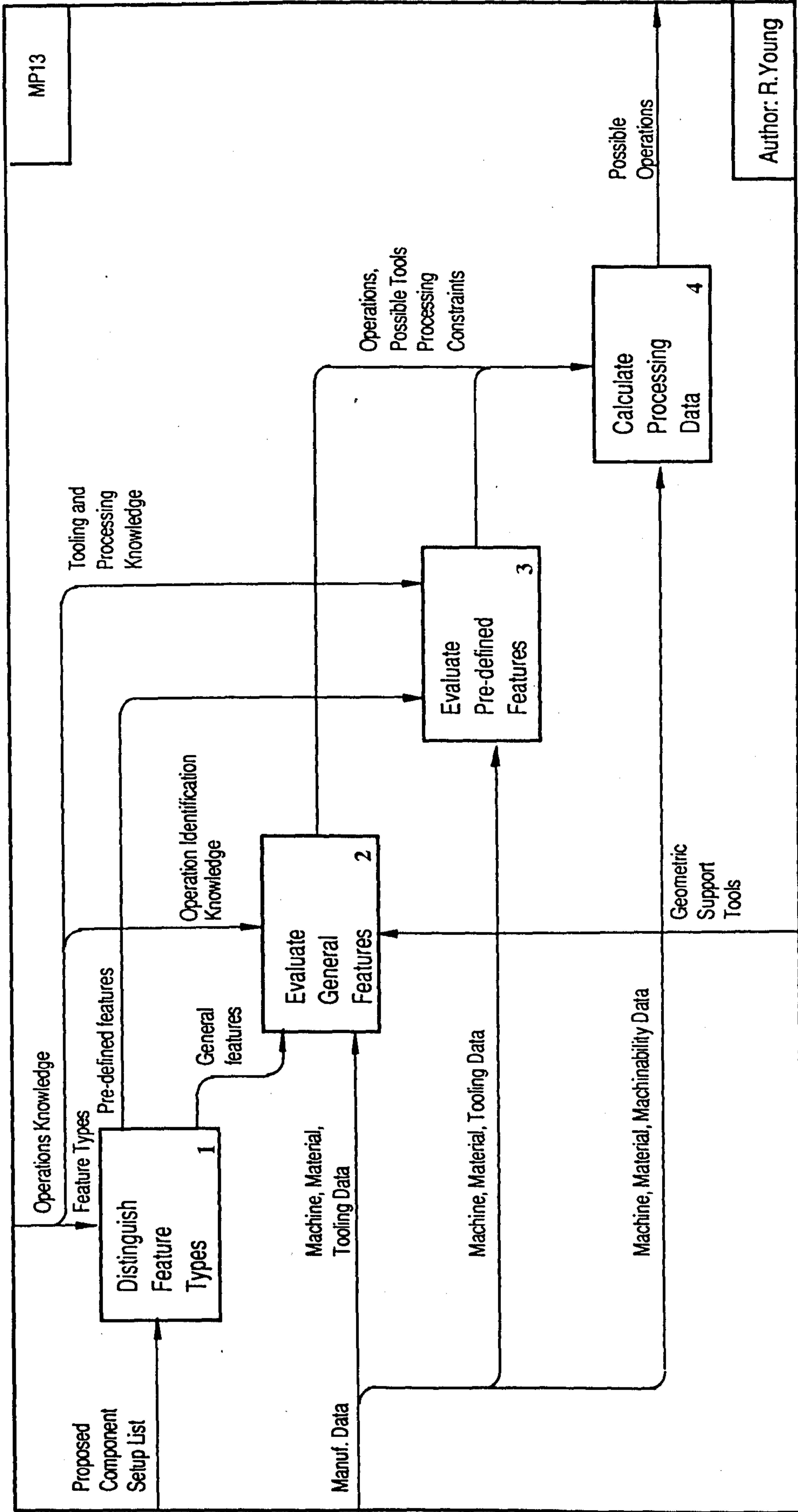
### Identify Features



Author: R. Young

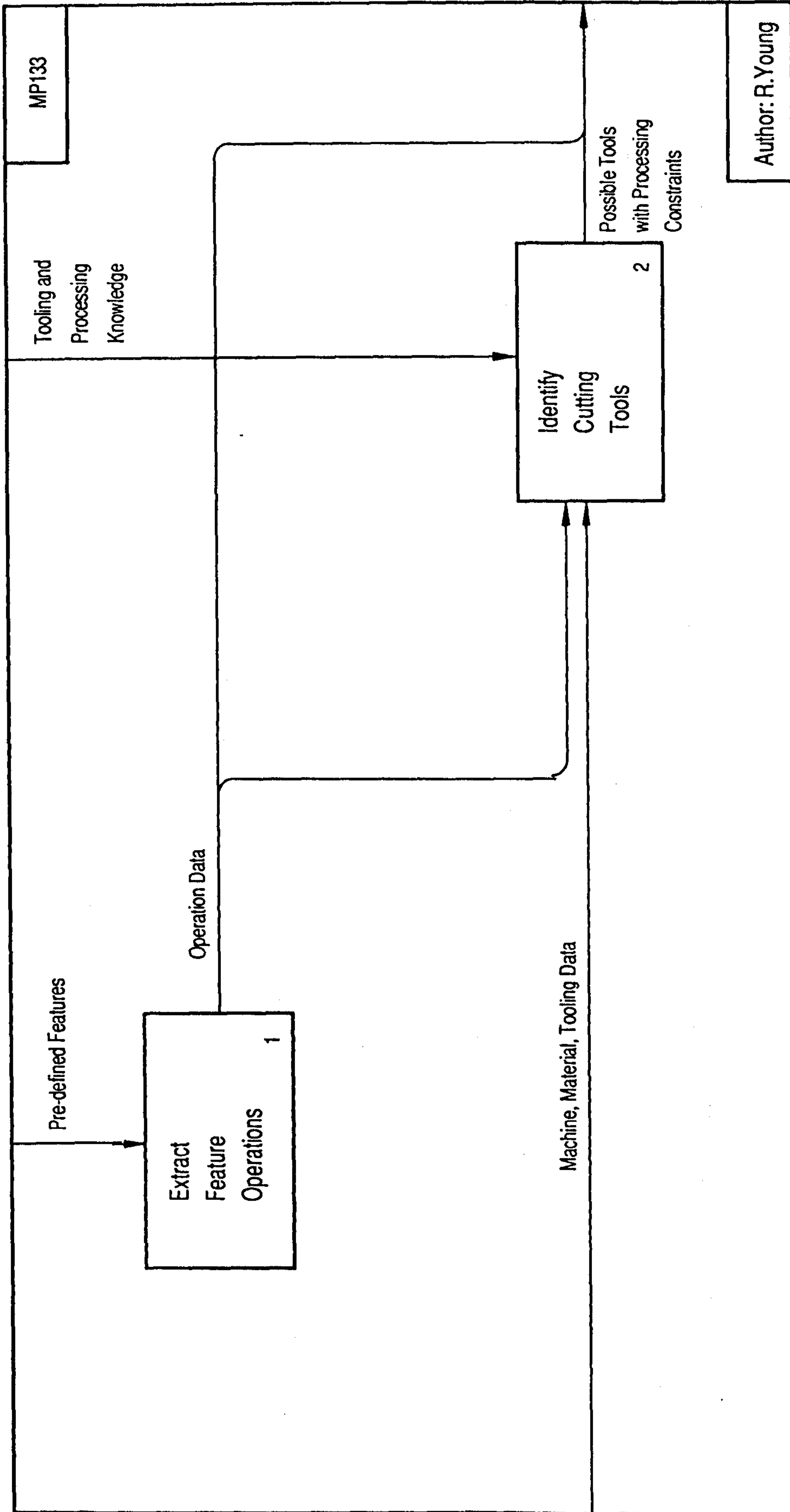
### Sequence Setups



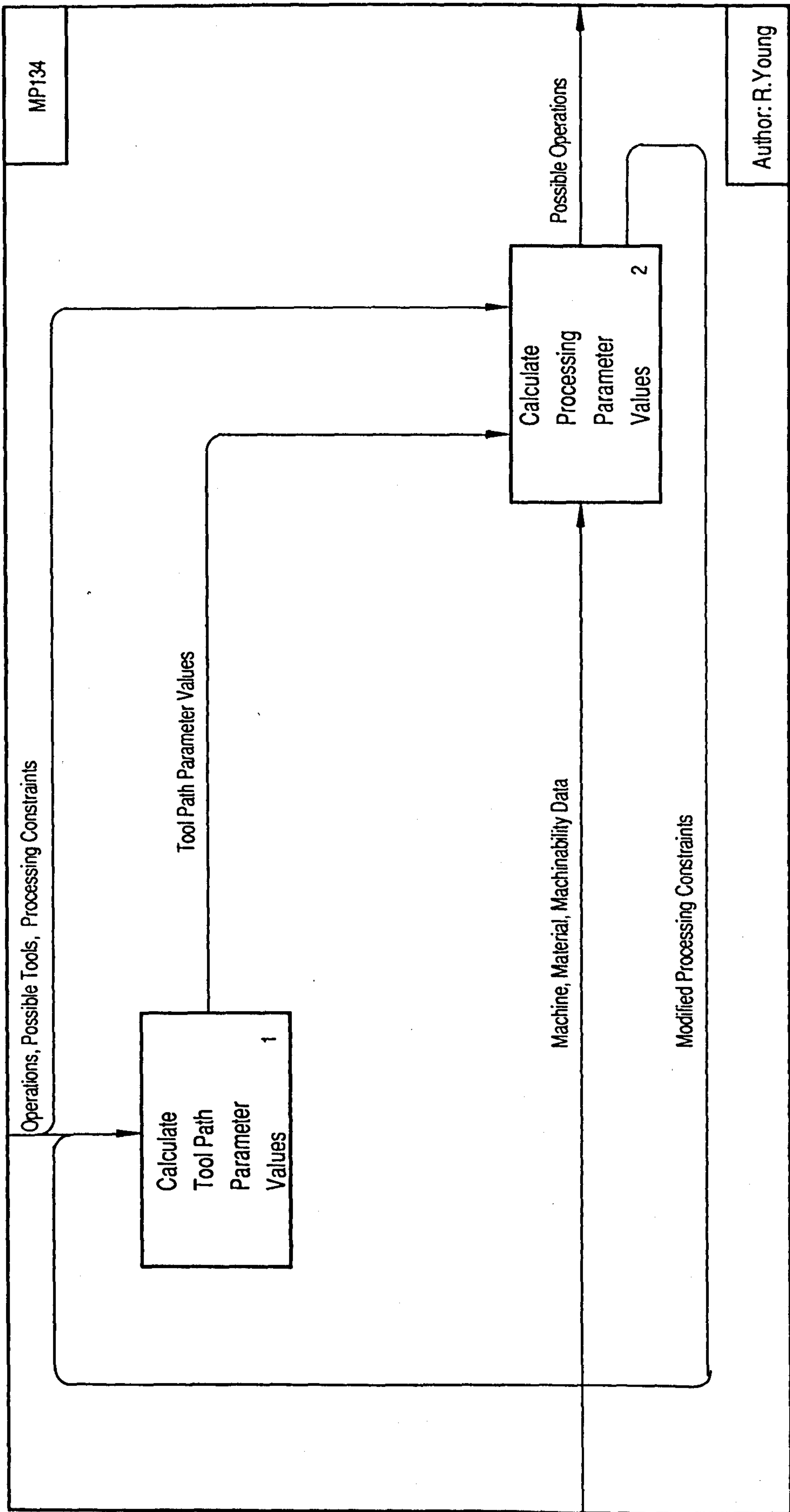


### Evaluate Features

Author: R. Young



### Evaluate Pre-defined Features



### Calculate Processing Data



**MACHINE PLANNING FOR MATERIAL REMOVAL.**

**GLOSSARY OF TERMS.**

<b>DESCRIPTION</b>	<b>OUTPUT</b>	<b>INPUT</b>
<p><b>Component Features:</b> The break down of what has to be done on the machine into meaningful features. This will consist of a set of faces, with tool approach directions, spindle axes, and fixture face normals. Also an identified processing method.</p>	<b>MP11</b>	<b>MP121</b>
<p><b>Component Setup Knowledge:</b> Information about what the company knows about Component Setups.</p>		<b>MP1</b>
<p><b>Component Setup Sequence:</b> A sequence of Setups for a component detailing the features to be machined, Operations to be performed, Possible Tools and Processing data for each Op/tool, Features for location and clamping.</p>	<b>MP14</b>	<b>MP2</b>
<p><b>Component Setup Sequence Knowledge:</b> Information about what the company knows about Component Setup Sequencing.</p>		<b>MP123,MP124</b>
<p><b>Expected Process Data:</b> Data indicating how the process is expected to proceed. e.g. Tool Life Utilisation, Machine Power required.</p>	<b>MP3</b>	
<p><b>Pre-defined Features:</b> machining features which come from parametric Pre-defined data.</p>	<b>MP131</b>	<b>MP1331</b>
<p><b>General Features:</b> machining feature information generated from analysis of the machine requirement.</p>	<b>MP131</b>	<b>MP132</b>
<p><b>Geometric Relationships:</b> The geometric relationships of features within a setup. This includes the position of a feature within a setup, the relationships between features in a setup, and the proximity of other features from any one feature.</p>	<b>MP122</b>	<b>MP123</b>

<b>Geometric Support Tools:</b> Tools which allow the computation of answers to geometric queries.		MP11,MP122 MP132,MP14 MP2
<b>Initial Component Setup List:</b> The first listing of groups of features.	MP121	MP122,MP123
<b>Machinability Data:</b> Data which is used to calculate speed and feed rate information.		MP1342
<b>Machine Batch Setup Knowledge:</b> Information about what the company knows about Machine Batch Setups. i.e. how to fixture batches of components onto specific machine types.		MP2
<b>Machine Batch Setup List:</b> A sequence of setups for the machine/component combination detailing the fixtures to be used, the features to be machined, Operations to be performed, Possible Tools and Processing data for each Op/tool, and features for location and clamping.	MP2	MP3
<b>Machine Data:</b> Data concerning the machine being considered, e.g. machine table size, speed/power graph, size of tool magazine, etc.		MP123,MP124 MP132,MP133 MP1342
<b>Machine Knowledge:</b> Information about what the company knows about how a Machine should be used.		MP
<b>Machine Plan:</b> Planning information detailing how to do what has been requested on a particular machine.	MP	
<b>Machine Requirement:</b> The description of what is needed of a particular machine. e.g. the pre- and post machined component descriptions, and the batch size.		MP11,MP122
<b>Machine Type:</b> The general description of the type of machine being considered.		MP1,MP2
<b>Manufacturing Data:</b> Data about the manufacturing environment. e.g. Machine data, tooling, fixturing, machinability, material data.		MP12,MP2
<b>Material Data:</b> Data describing the properties of the material being used to produce a component.		MP12, MP132 MP1332,MP13

<p><b>Modified Component Setup List:</b> A component Setup List which has been changed, e.g. when tool accessibility problems are identified.</p>	MP14	MP13
<p><b>Modified Processing Constraints:</b> A changed processing constraint value. e.g. reduction in max. depth of cut due to machine power limitations.</p>	MP1342	MP1341
<p><b>Operations,Possible Tools, Processing Constraints:</b> a list of Operations, each of which has associated with it the tools which can be considered, and the constraints on how the tools can be used.</p>	MP132 MP133	MP134
<p><b>Operation Data:</b> Data which describes a tool specific piece of machining.</p>	MP1331	MP1341,MP13
<p><b>Operation Identification Knowledge:</b> Information about what the company knows about how regions can be broken down to provide Operation data.</p>		MP132
<p><b>Operation Knowledge:</b> Information about what the company knows about how features can be processed.</p>		MP13
<p><b>Operation List:</b> A sequenced list of Operations, with specific tools, for a setup.</p>	MP3	MP4
<p><b>Part Program Requirement, Machine and Controller Identification:</b> The identification of the specific machine and its controller to be used in the production of a component.</p>	MSG32	MP4
<p><b>Part Programs:</b> Generated NC code which has still to be proven.</p>	MP4	
<p><b>Possible Operations:</b> All data about how an Operation may be performed.</p>	MP1342	MP14
<p><b>Possible Tools with Processing Constraints:</b> A list of tools which can be considered in performing an Operation, with the constraints on how the tools can be used.</p>	MP1332	MP1341,MP13



<b>Preliminary Setup List:</b> The status of the Component Setup List once any setups which need to be done before the principal setup have been identified.	<b>MP124</b>	<b>MP125</b>
<b>Principal Setup List:</b> The status of the Component Setup List once the principal setup has been identified.	<b>MP123</b>	<b>MP124</b>
<b>Proposed Component Setup List:</b> The status of the Component Setup List before features have been evaluated.	<b>MP12</b>	<b>MP13,MP14</b>
<b>Feature Identification Knowledge:</b> Information about what the company knows about how components can be broken down into features.		<b>MP11</b>
<b>Feature Types:</b> The types of feature which can be used, i.e. Pre-defined and General Features.		<b>MP131</b>
<b>Sequencing Knowledge:</b> Information about what the company knows about how a list of operations should be sequenced.		<b>MP3</b>
<b>Tooling &amp; Processing Knowledge:</b> Information about what the company knows about how to select tools and processing constraints.		<b>MP1332</b>
<b>Tooling Data:</b> Data specifying the tools available for use on a specific machine type.		<b>MP132,MP133</b>
<b>Tool Path Parameter Values:</b> Values of path length, cut depth and width for an Operation/tool combination.	<b>MP1341</b>	<b>MP1342</b>

## APPENDIX 4

## DATA REPRESENTATIONS BUILT IN THE RESEARCH

## 4.1 INTRODUCTION

This appendix provides representations of machine tool data, cutter data, feature and component definitions from which the implemented Closely Coupled Machine Planner has taken its data.

The machine and cutter data representation are taken from the first version of the PDM, "feat.lam". These have been reinstated in subsequent versions as an adequate base for the exploratory software. The feature representations are provided from the "machplan.lam" version of the PDM, with the exception of the channel definition in section 4.4.1 which is drawn from the "pms3\_8.lam" version of the PDM. The variant of the Bolster Plate definition is taken from "machplan.lam", while the Glacier reduced size bearing definition is taken from the "pms3\_8.lam" version of the PDM, which drew together the key elements of data required for all the MCG Applications.

## 4.2 MACHINE DATA REPRESENTATION

The following machine data representation has been populated with data related to the Wadkin V4-6 machining centre which has been used in the experimental machining work.

2474 mc-tool-database:

2475 Reference Number: 1 Description: "wadkin V4/6"

2476 Magazine:

2477 Pocket number: 1 - Actual tool number: x  
 2478 Pocket number: 2 - Actual tool number: x  
 2479 Pocket number: 3 - Actual tool number: x  
 2480 Pocket number: 4 - Actual tool number: x  
 2481 Pocket number: 5 - Actual tool number: x  
 2482 Pocket number: 5 - Actual tool number: x  
 2483 Pocket number: 6 - Actual tool number: x  
 2484 Pocket number: 6 - Actual tool number: x  
 2485 Pocket number: 7 - Actual tool number: x  
 2486 Pocket number: 8 - Actual tool number: x  
 2487 Pocket number: 9 - Actual tool number: x  
 2488 Pocket number: 10 - Actual tool number: x  
 2489 Pocket number: 11 - Actual tool number: x  
 2490 Pocket number: 12 - Actual tool number: x  
 2491 Pocket number: 13 - Actual tool number: x  
 2492 Pocket number: 14 - Actual tool number: x  
 2493 Pocket number: 15 - Actual tool number: x  
 2494 Pocket number: 16 - Actual tool number: x  
 2495 Pocket number: 17 - Actual tool number: x  
 2496 Pocket number: 18 - Actual tool number: x  
 2497 Pocket number: 19 - Actual tool number: x  
 2498 Pocket number: 20 - Actual tool number: x  
 2499 Pocket number: 21 - Actual tool number: x  
 2500 Pocket number: 22 - Actual tool number: x  
 2501 Pocket number: 23 - Actual tool number: x  
 2502 Pocket number: 24 - Actual tool number: x  
 2503 Pocket number: 25 - Actual tool number: x  
 2504 Pocket number: 26 - Actual tool number: x

2505 Pocket number: 27 - Actual tool number: x  
 2506 Pocket number: 28 - Actual tool number: x  
 2507 Pocket number: 29 - Actual tool number: x  
 2508 Pocket number: 30 - Actual tool number: x  
 2509 Machine spec:  
 2510 Description : "wadkin V4/6"  
 2511 Machine type : machining centre  
 2512 Maximum part length : 450.0  
 2513 Maximum part width : 300.0  
 2514 Maximum part depth : 400.0  
 2515 Minimum surface finish : 0.8  
 2516 Minimum tolerance : 0.025  
 2517 Minimum speed : 20.0  
 2518 Speed characteristic - point 1 : 2000.0  
 2519 Speed characteristic - point 2 : 4000.0  
 2520 Maximum speed : 10000.0  
 2521 Power constants :C1 = 0.005966, C2 = 11.5, C3 = 14.3000002, C4 = 0.0007  
 2522 Maximum power : 11.5  
 2523 Maximum torque : 57.0  
 2524 Machining rate : 35.0  
 2525 Tool change time : 0.17  
 2526 Tool magazine change time : 60.0  
 2527 Rigidity factor : 0.85  
 2528  
 2529

### 4.3 CUTTER DATA REPRESENTATION

The data representation shown below is taken from the tooling database representation in the PDM and represents the structure of data, for one tool, constructed by A. Mckay [93]. The tooling data required for Machine Planning was extracted from this representation.

1845 x x x x x: Base diameter = 63.0 mm Corner rad = 0.8 mm  
 1846 Base angle = 15.0 degrees Side angle = 15.0 degrees  
 1847 7.0 edges 10000.0 rpm (max)  
 1848  
 1849 Life used = 0.00000E+00% Holder assy:  
 1850 Holding method of tool : 0  
 1851 Machine tool holder : 0.00000E+00 0  
 1852  
 1853 "0"  
 1854 Flute length = 9.0 mm Gauge length = 96.0 mm  
 1855 True radius = 1.0 mm Dog angle = 96.0 degrees  
 1856 Tip details :  
 1857 tip details code : x  
 1858 Geometry Feature 1 = 0.00000E+00 , Geometry Feature  
 2 = 0.00000E+00  
 1859 Geometry Feature 3 = 0.00000E+00 , Geometry Feature  
 4 = 0.00000E+00  
 1860 Geometry Feature 5 = 0.00000E+00 , Geometry Feature  
 6 = 0.00000E+00  
 1861 Cost per tooth = 2.1300001 pounds  
 1862 Norm Clearance = x , Tolerance class = x  
 1863 Cutting Edge Cond = x , Fixing and breakers = x  
 1864 Tip ISO code : SCMM090408TR  
 1865 Tip material :Tip material spec : k10st , code number :  
 x  
 1866 Max chip thicknesses :  
 1867  
 1868 Maximum chip thickness = 0.15 mm  
 1869 TC1 = 9.95000E+06 \* 10\*\*6



1870 TC2 = 3.0 , TC3 = 2.8  
 1871 TC4 = 1.3 , TC5 = 1.0 , TC6 = 0.7  
 1872 Chip Reference :  
 1873 Material : grade 17 , Spec : grade 17 , Condition : x  
 1874 Hardness range : 190.0 - 250.0 Bhn  
 1878 Rigidity factor = 1.0

#### 4.4 FEATURE REPRESENTATIONS

The first feature representation in the following list is taken from the "pms3\_8.lam" version of the PDM, which has been argued in this thesis. The remaining features, represented below, have been taken from the "machplan.lam" version of the PDM, discussed in chapter 10.6. These contain the same range of data, apart from the bounding box, but it is organised differently and contains a number of data fields which are not required by the Machine Planner. Three features, the Channel, the Face and the Hole have been implemented in "pms3\_8.lam" the version of the PDM, and tested using the Glacier workpiece, used in the experiment described in chapter 11.4.

##### 4.4.1 The "pms3\_8" Channel Representation

The feature definition data for the Channel feature using the final version of the PDM. The horizontal lines across the page represent a level change, where the data following the line provides a break down of one aspect of the data from above the line.

```
-----
{feature machining data} -- Collection
pre-machined feature : ?? : ??
relationship graph : coordinate relationship: <zoom in>
machine class : mill
machining approach directions :
  +x : impossible
  -x : ??
  +y : impossible
  -y : ??
  +z : possible
  -z : ??
fixture face normals :
  +x : impossible
  -x : ??
  +y : possible
  -y : possible
  +z : possible
  -z : ??
bounding box :
  +x : half ( {20} )
  -x :
    ?? : ( - half ( {20} ) )
  +y : half ( {23} )
  -y :
    ?? : ( - half ( {23} ) )
  +z :
    zero : ( 0.00000E+00 mm )
  -z :
    ?? : ( - {29 depth} )
volume :
  x * y * z : {23 width} * {29 depth} * {20 length}
operation activity : sequential set :
  OPERATION ACTIVITY:
```

name : 1  
 basic activity data : <zoom in>  
 comment : "rough cut"

OPERATION ACTIVITY:

name : 2  
 basic activity data : <zoom in>  
 comment : "finish cut"

post-machined feature : ?? : ??

OPERATION ACTIVITY:

name : 1  
 basic activity data : <zoom in>  
 comment : "rough cut"  
 {operation data} -- Collection  
 primary regions : ...  
 secondary regions : ...  
 relationship graph : ...  
 operation definition : ...  
 constraints : ...  
 geometric information : ...  
 tooling : ...

{operation definition} -- Collection  
 operation type : composite op ( open slot mill , end mill )  
 cut type : rough  
 tool path plan : clear\_channel

{constraints} -- Collection  
 tool approach directions : ?? : ??  
 ?? : ??

dimensional tolerance :  
 ?? : ??  
 ?? : ??

surface roughness :  
 ?? : ??  
 ?? : ??

diameter constraints :  
 width - 2 \* finish cut : ( {23 width} -  
 two \* finish cut : ... )

corner radii constraints :  
 ?? : ??  
 ?? : ??

length constraints : {29 depth}

{geometric information} -- Collection  
 operation length : {20 length}  
 operation width :  
 width - 2 \* finish cut : ( {23 width} -  
 two \* finish cut : ( two \* finish cut ) )  
 operation depth :  
 depth - finish cut : ( {29 depth} - finish cut )  
 operation side width :  
 ?? : ??  
 operation side depth :  
 ?? : ??  
 stock width :  
 width - 2 \* finish cut : ( {23 width} -  
 two \* finish cut : ( two \* finish cut ) )  
 corner radius :  
 ?? : ??

{tooling} -- Collection  
 possible tools : ...  
 ...  
 chosen tool :  
 relationship graph : ...  
 actual tool : ...  
 subops : ...  
 tool plan : ??  
 maximum cut depth : ...  
 minimum cut depth : ...  
 cutting method : ??  
 tool offset : ...

-----  
 {subops} -- List

feed rate :  
 ?? : ??  
 speed :  
 ?? : ??  
 power :  
 ?? : ??  
 cut depth :  
 ?? : ??  
 cut width :  
 ?? : ??  
 path length :  
 ?? : ??

-----  
 OPERATION ACTIVITY:

name : 2  
 basic activity data : <zoom in>  
 comment : "finish cut"

-----  
 {operation definition} -- Collection

operation type : end mill  
 cut type : finish face and side  
 tool path plan : face\_channel

-----  
 {constraints} -- Collection

tool approach directions : ?? : ??  
 ?? : ??  
 dimensional tolerance : {23 width}  
 {29 depth}  
 {20 length}  
 surface roughness : surface finish  
 diameter constraints :  
 width - 2 \* finish cut : ( {23 width} -  
 two \* finish cut : ( ... \* ... ) )  
 corner radii constraints :  
 ?? : ??  
 ?? : ??  
 length constraints : {29 depth}

-----  
 {geometric information} -- Collection

operation length : {20 length}  
 operation width : finish cut  
 operation depth : finish cut  
 operation side width : {23 width}  
 operation side depth : {29 depth}  
 stock width : {23 width}  
 corner radius :  
 ?? : ??

-----

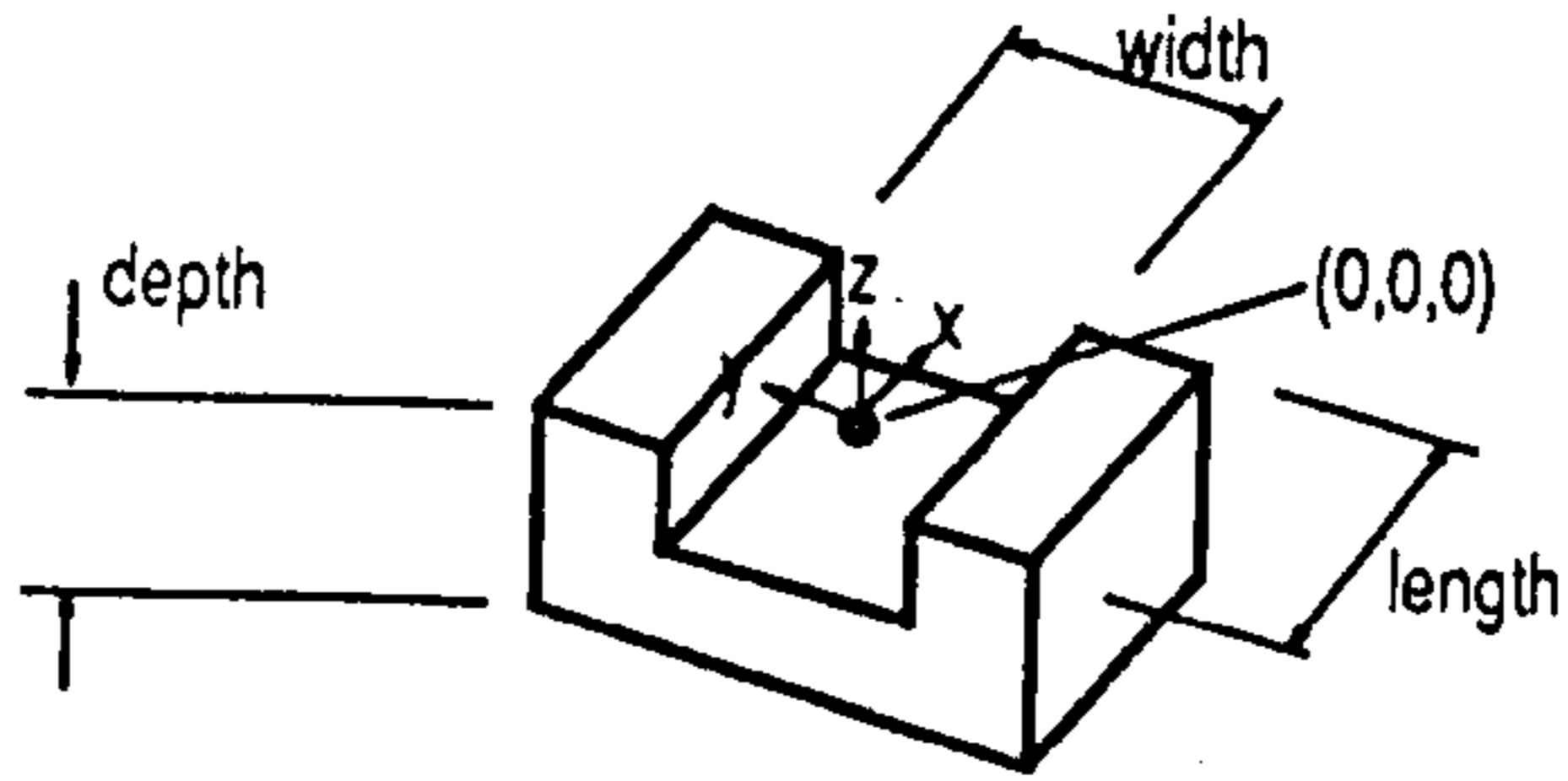


4.4.2 The Channel

```

thru_channel == (
  width == 50.0,
  length == 40.0,
  depth == 20.0,
  surface finish == 2.5,
  xdatum == 0.00000E+00,
  ydatum == 0.00000E+00,
  zdatum == 0.00000E+00) =
Null

```



```

Intersection:
+x face : Transformed solid:
  Plane
  AT
  ry = 90.0,mx = length / 2.0
-x face : Transformed solid:
  Plane
  AT
  ry =- ( 90.0 ),mx =- ( length / 2.0 )
+y face : Transformed solid:
  Plane
  AT
  rx =- ( 90.0 ),my = width / 2.0
-y face : Transformed solid:
  Plane
  AT
  rx = 90.0,my =- ( width / 2.0 )
+z face : Plane
-z face : Transformed solid:
  Plane
  AT
  rx = 180.0,mz =- ( depth )

```

```

machining data
reg_constraints:
  spindle_axes:
    minus_z
  tool_app_dims:
    minus_z
    plus_x
    minus_x
  fix_face_norms:
    plus_y
    minus_y
    plus_z
  region volume:
    length * width * depth
  region length: length
  region width: width
  region depth: depth
operations:
  machining method pre_defined
  basic op data
  current cut roughing
  approach dim ??
  machining element
  element number 6.0
  element name clear_channel
  constraints:
    operation tolerance

```

```

    ??
    surface roughness
    ??
    diameter constraints
    ( width -
      2.0 * finish allow
    )
    radius constraints
    ??
    length constraints
    depth
op type data
current op
  open_slotting
  end_milling

width ( width -
  finish allow * 2.0
)
depth ( depth - finish allow)
length length
radius ??
stock width ( width -
  finish allow * 2.0
)
width2 ??
depth2 ??

machining method pre_defined
basic op data
  current cut fin_facenside
  approach dirn ??
  machining element
  element number 7.0
  element name face_channel
constraints:
  operation tolerance
  width
  length
  depth
  xdatum
  ydatum
  zdatum
  surface roughness
  ??
  diameter constraints
  ( width -
    2.0 * finish allow
  )
  radius constraints
  ??
  length constraints
  ??
op type data
  current op end_milling
  width finish allow
  depth finish allow
  length length
  radius ??
  stock width finish allow
  width2 width
  depth2 depth

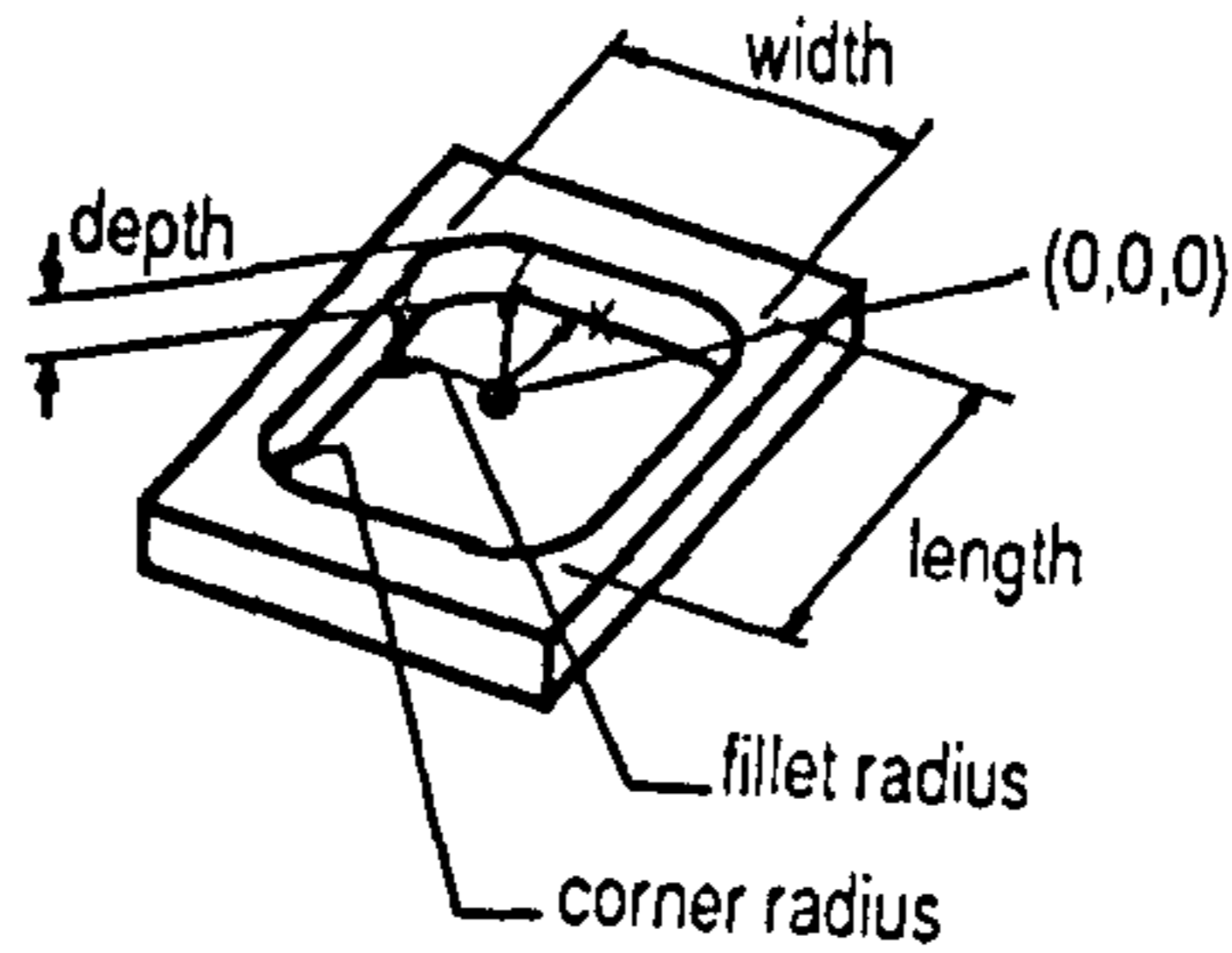
```

"channel length in x, (0,0,0) at mid x,y, top z

4.4.3 The Closed Pocket

```

moved clsd pocke == (
  width == ?? ,
  length == ?? ,
  depth == ?? ,
  corner rad == ?? ,
  fillet rad == ?? ,
  surface finish == ?? ,
  xdatum == ?? ,
  ydatum == ?? ,
  zdatum == ?? ) =
    
```



Null

Intersection:

Plane

Transformed solid:

Plane

AT

rx = 180.0,mz =- ( depth )

Union:

Union:

narrow-one : Intersection:

Transformed solid:

Plane

AT

rx = -90.0,

my = ( width / 2.0 - corner rad)

Transformed solid:

Plane

AT

rx = 90.0,

my =- (( width / 2.0 - corner rad) )

Transformed solid:

Plane

AT

ry = -90.0,mx =- ( length / 2.0 )

Transformed solid:

Plane

AT

ry = 90.0,mx = length / 2.0

short-one : Intersection:

Transformed solid:

Plane

AT

rx = -90.0,my = width / 2.0

Transformed solid:

Plane

AT

rx = 90.0,my =- ( width / 2.0 )

Transformed solid:

Plane

AT

ry = -90.0,

mx =- (( length / 2.0 - corner rad) )

Transformed solid:

Plane

AT

ry = 90.0,



$$mx = (length / 2.0 - corner\ rad)$$

Union:

four-corners : Union:

top-left-hand : Transformed solid:

Cylinder:Radius = corner rad

AT

mx =- (( length / 2.0 - corner rad) ),

my =( width / 2.0 - corner rad)

top-right-hand : Transformed solid:

Cylinder:Radius = corner rad

AT

mx =( length / 2.0 - corner rad),

my =( width / 2.0 - corner rad)

bot-right-hand : Transformed solid:

Cylinder:Radius = corner rad

AT

mx =( length / 2.0 - corner rad),

my =- (( width / 2.0 - corner rad) )

bot-left-hand : Transformed solid:

Cylinder:Radius = corner rad

AT

mx =- (( length / 2.0 - corner rad) ),

my =- (( width / 2.0 - corner rad) )

machining data

reg\_constraints:

spindle\_axes:

minus\_z

tool\_app\_dirs:

minus\_z

fix\_face\_norms:

plus\_z

plus\_x

minus\_x

plus\_y

minus\_y

region volume:

depth \*(

length \* width

- (

4.0 \* corner rad \* corner rad

-

3.1400001 \* corner rad \* corner rad

))

region length: length

region width: width

region depth: depth

operations:

machining method pre\_defined

basic op data

current cut roughing

approach dim ??

machining element

element number 1.0

element name clear\_pocket

constraints:

operation tolerance

??

surface roughness

??

diameter constraints

( length -

2.0 \* finish allow

```

    )
    ( width -
      2.0 * finish allow
    )
    4.0 * corner rad
radius constraints
??
length constraints
depth
op type data
current op
  closed_slotting
  open_slotting
  end_milling
width ( width -
  1.0 * corner rad
)
depth depth
length ( length -
  1.0 * corner rad
)
radius 0.00000E+00
stock width ( width -
  1.0 * corner rad
)
width2 ??
depth2 ??

machining method pre_defined
basic op data
current cut roughing
approach dirn ??
machining element
  element number 2.0
  element name profile_pocket
constraints:
  operation tolerance
  ??
  surface roughness
  ??
  diameter constraints
  1.8 *( corner rad - finish allow)
  radius constraints
  ??
  length constraints
  depth
op type data
current op end_milling
width ( width -
  2.0 * finish allow
)
depth depth
length ( length -
  2.0 * finish allow
)
radius ( corner rad - finish allow)
stock width
      ( corner rad / 2.0 - finish allow)
width2 ??
depth2 ??

machining method pre_defined
basic op data

```

```

current cut finishing
approach dirn ??
machining element
  element number 2.0
  element name profile_pocket
constraints:
  operation tolerance
  width
  length
  surface roughness
  surface finish
  diameter constraints
    1.8 * corner rad
  radius constraints
    fillet rad
  length constraints
  depth
op type data
  current op end_milling
  width width
  depth depth
  length length
  radius corner rad
  stock width finish allow
  width2 ??
  depth2 ??

```

"Pocket with axes and top surf at (0,0,0)

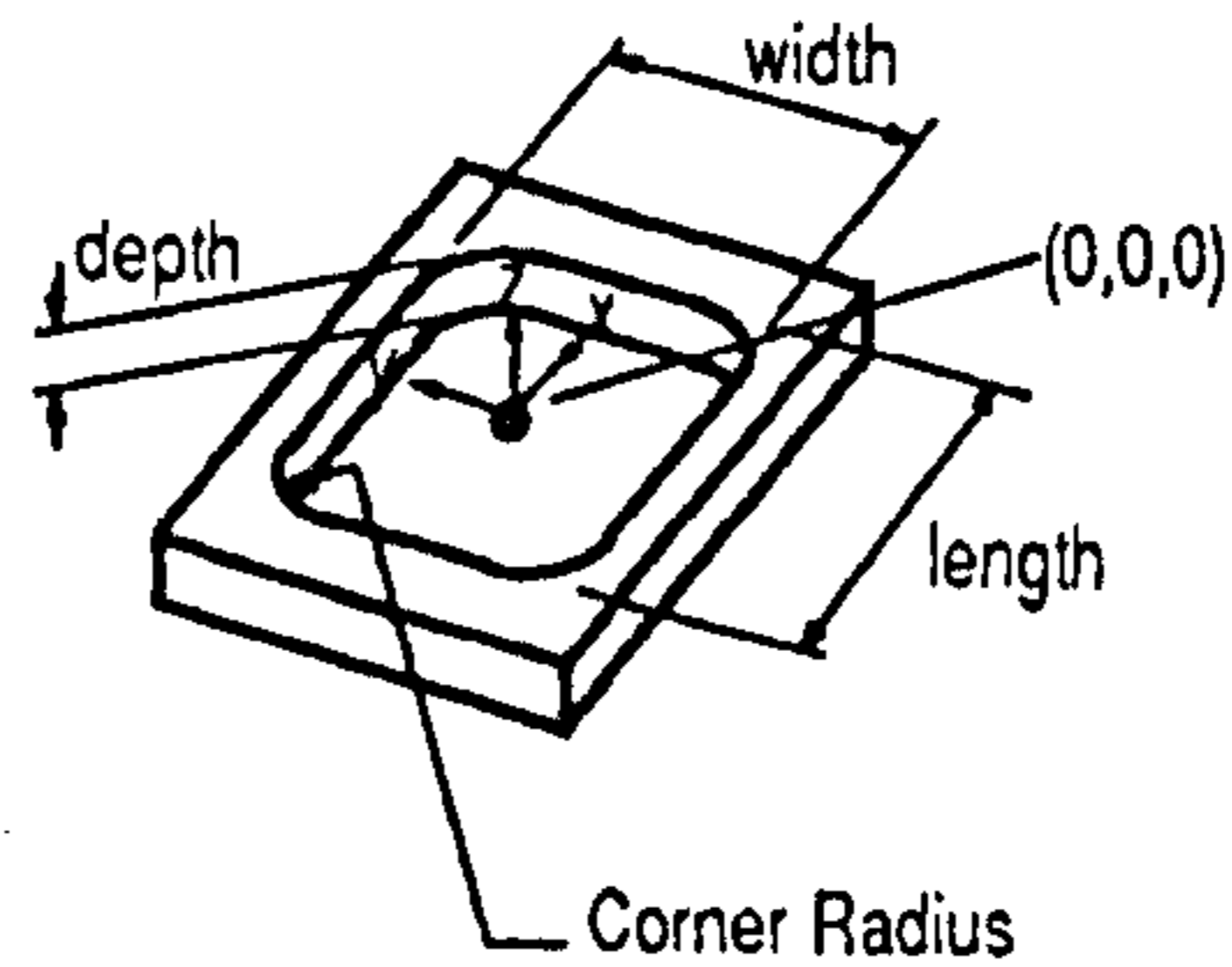
#### 4.4.4 The Through Pocket

```

thru pocket == (
  width == ?? ,
  length == ?? ,
  depth == ?? ,
  corner rad == ?? ,
  surface finish == ?? ,
  xdatum == ?? ,
  ydatum == ?? ,
  zdatum == ?? ) =

```

Null



Intersection:

Plane

Transformed solid:

Plane

AT

rx = 180.0,mz = - ( depth )

Union:

Union:

narrow-one : Intersection:

Transformed solid:

Plane

AT

rx = -90.0,

my = ( width / 2.0 - corner rad)

Transformed solid:

Plane

AT

rx = 90.0,

my = - ( ( width / 2.0 - corner rad ) )

Transformed solid:

Plane

AT



```

        ry = -90.0, mx =- ( length / 2.0 )
Transformed solid:
  Plane
  AT
        ry = 90.0, mx = length / 2.0
short-one : Intersection:
Transformed solid:
  Plane
  AT
        rx = -90.0, my = width / 2.0
Transformed solid:
  Plane
  AT
        rx = 90.0, my =- ( width / 2.0 )
Transformed solid:
  Plane
  AT
        ry = -90.0,
        mx =- (( length / 2.0 - corner rad) )
Transformed solid:
  Plane
  AT
        ry = 90.0,
        mx =( length / 2.0 - corner rad)
Union:
four-corners : Union:
top-left-hand : Transformed solid:
  Cylinder:Radius = corner rad
  AT
        mx =- (( length / 2.0 - corner rad) ),
        my =( width / 2.0 - corner rad)
top-right-hand : Transformed solid:
  Cylinder:Radius = corner rad
  AT
        mx =( length / 2.0 - corner rad),
        my =( width / 2.0 - corner rad)
bot-right-hand : Transformed solid:
  Cylinder:Radius = corner rad
  AT
        mx =( length / 2.0 - corner rad),
        my =- (( width / 2.0 - corner rad) )
bot-left-hand : Transformed solid:
  Cylinder:Radius = corner rad
  AT
        mx =- (( length / 2.0 - corner rad) ),
        my =- (( width / 2.0 - corner rad) )

machining data
reg_constraints:
  spindle_axes:
    plus_z
    minus_z
  tool_app_dirs:
    plus_z
    minus_z
  fix_face_norms:
    plus_x
    minus_x
    plus_y
    minus_y
region volume:
  depth *(
    length * width

```

```

-
  0.86 * corner rad * corner rad
)
region length: length
region width: width
region depth: depth

```

operations:

```

machining method pre_defined
basic op data
  current cut roughing
  approach dim ??
  machining element
    element number 1.0
    element name clear_pocket
constraints:
  operation tolerance
    ??
  surface roughness
    ??
  diameter constraints
    ( length -
      2.0 * finish allow
    )
    ( width -
      2.0 * finish allow
    )
    4.0 * corner rad

  radius constraints
    ??
  length constraints
  depth
op type data
  current op
    closed_slotting
    open_slotting
    end_milling
  width ( width -
    1.0 * corner rad
  )
  depth
    depth + 3.0

  length ( length -
    1.0 * corner rad
  )
  radius 0.00000E+00
  stock width ( width -
    1.0 * corner rad
  )
  width2 ??
  depth2 ??

```

```

machining method pre_defined
basic op data
  current cut roughing
  approach dim ??
  machining element
    element number 2.0
    element name profile_pocket
constraints:

```

```

operation tolerance
  ??
surface roughness
  ??
diameter constraints
  1.8 *( corner rad - finish allow)
radius constraints
  ??
length constraints
  depth
op type data
current op end_milling
width ( width -
  2.0 * finish allow
)
depth
  depth + 3.0

length ( length -
  2.0 * finish allow
)
radius ( corner rad - finish allow)
stock width
  ( corner rad / 2.0 - finish allow)
width2 ??
depth2 ??

machining method pre_defined
basic op data
current cut finishing
approach dim ??
machining element
  element number 2.0
  element name profile_pocket
constraints:
  operation tolerance
  width
  length
  surface roughness
  surface finish
  diameter constraints
    1.8 * corner rad
  radius constraints
    ??
  length constraints
  depth
op type data
current op end_milling
width width
depth
  depth + 3.0
length length
radius corner rad
stock width finish allow
width2 ??
depth2 ??

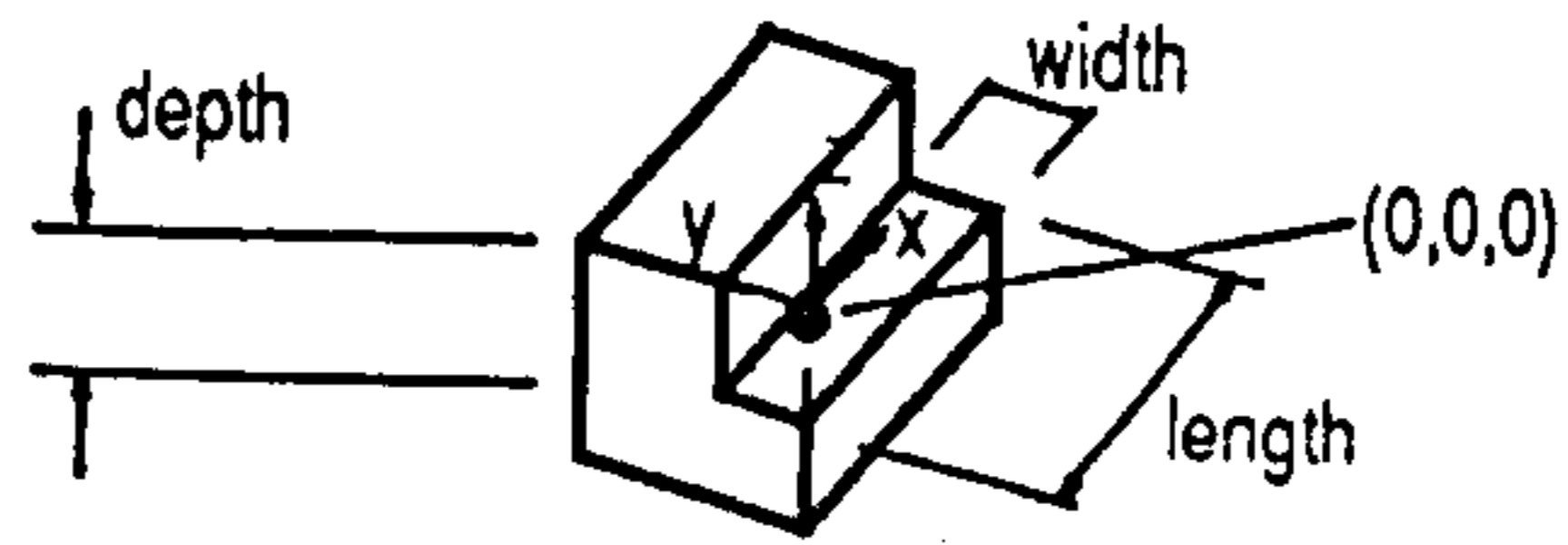
```

"Thru pocket defined with axes and top surface at (0,0,0)



4.4.5 The Step

```
stepped edge == (
  width == 50.0,
  length == 40.0,
  depth == 20.0,
  surface finish == 2.5,
  fillet rad == ?? ,
  xdatum == 0.00000E+00,
  ydatum == 0.00000E+00,
  zdatum == 0.00000E+00) =
Null
```



Intersection:

```
+x face : Transformed solid:
  Plane
  AT
  ry = 90.0,mx = length
-x face : Transformed solid:
  Plane
  AT
  ry =- ( 90.0 )
+y face : Transformed solid:
  Plane
  AT
  rx =- ( 90.0 ),my = width
-y face : Transformed solid:
  Plane
  AT
  rx = 90.0
+z face : Plane

-z face : Transformed solid:
  Plane
  AT
  rx = 180.0,mz =- ( depth )
```

machining data

```
reg_constraints:
  spindle_axes:
    minus_z
    plus_y
  tool_app_dirs:
    minus_z
    plus_y
    plus_x
    minus_x
  fix_face_norms:
    plus_z
    minus_y
  region volume:
    length * width * depth
  region length: length
  region width: width
  region depth: depth
operations:
  machining method pre_defined
  basic op data
  current cut roughing
  approach dirn ??
  machining element
  element number 3.0
  element name clear step
  constraints:
```

```

operation tolerance
  ??
surface roughness
  ??
diameter constraints
  ??
radius constraints
  ??
length constraints
  depth
op type data
current op end_milling
width ( width - finish allow)
depth ( depth - finish allow)
length length
radius ??
stock width ( width - finish allow)
width2 ??
depth2 ??

machining method pre_defined
basic op data
current cut fin_facenside
approach dirn ??
machining element
  element number 4.0
  element name finish step
constraints:
  operation tolerance
    width
    length
    depth
    xdatum
    ydatum
    zdatum
  surface roughness
  surface finish
  diameter constraints
    ??
  radius constraints
    fillet rad
  length constraints
    depth
op type data
current op end_milling
width finish allow
depth finish allow
length length
radius ??
stock width finish allow
width2 width
depth2 depth

```

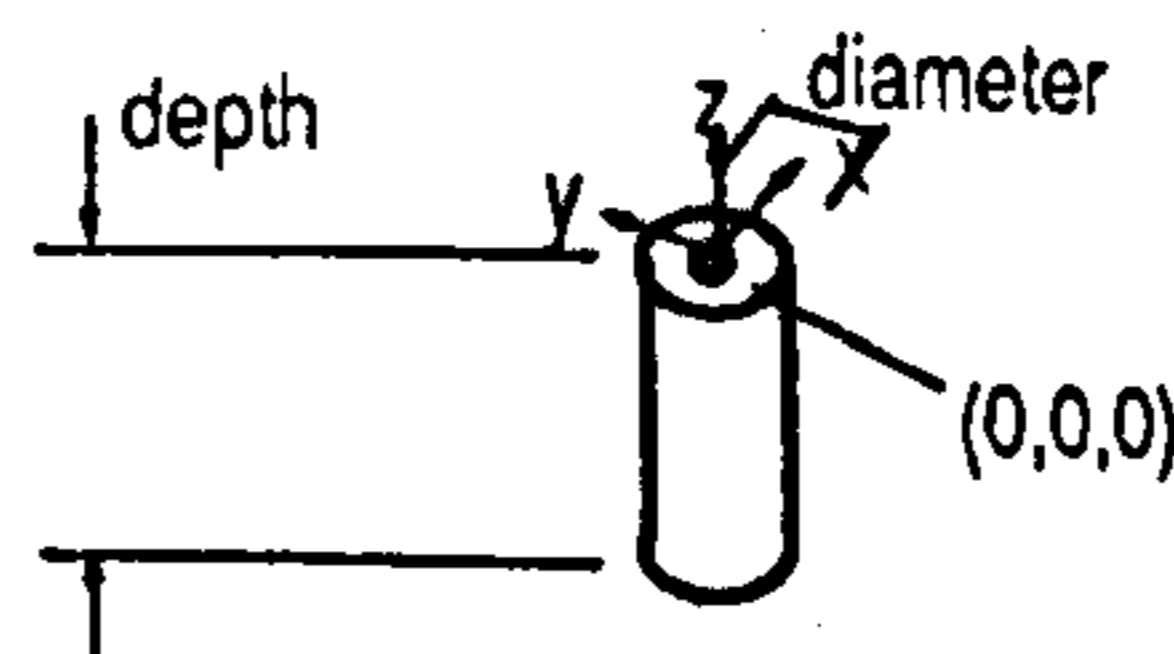
"step length in x,(0,0,0) at front,left,top

#### 4.4.6 The Hole

```

simp_thru_hole == (
  diameter == dia,
  depth == depth,
  sfin == ?? ,
  xdatum == ?? ,
  ydatum == ?? ,

```



```

zdatum == ?? ) =
Null

Cylinder:Radius = diameter / 2.0
machining data
reg_constraints:
  spindle_axes:
    plus_z
    minus_z
  tool_app_dirs:
    plus_z
    minus_z
  fix_face_norms:
    all_xy
  region volume:
    depth *
      3.1400001 * diameter * diameter
      / 4.0
  region length: depth
  region width: diameter
  region depth: depth
operations:
  machining method pre_defined
  basic op data
  current cut roughing
  approach dirn ??
  machining element
  element number 8.0
  element name drill_hole
  constraints:
    operation tolerance
    diameter
    surface roughness sfin
    diameter constraints
    diameter
    radius constraints
    ??
    length constraints
    depth
  op type data
  current op drilling
  width diameter / 2.0
  depth
  depth + 7.0
  length
  depth + 7.0
  radius diameter / 2.0
  stock width diameter
  width2 ??
  depth2 ??

```

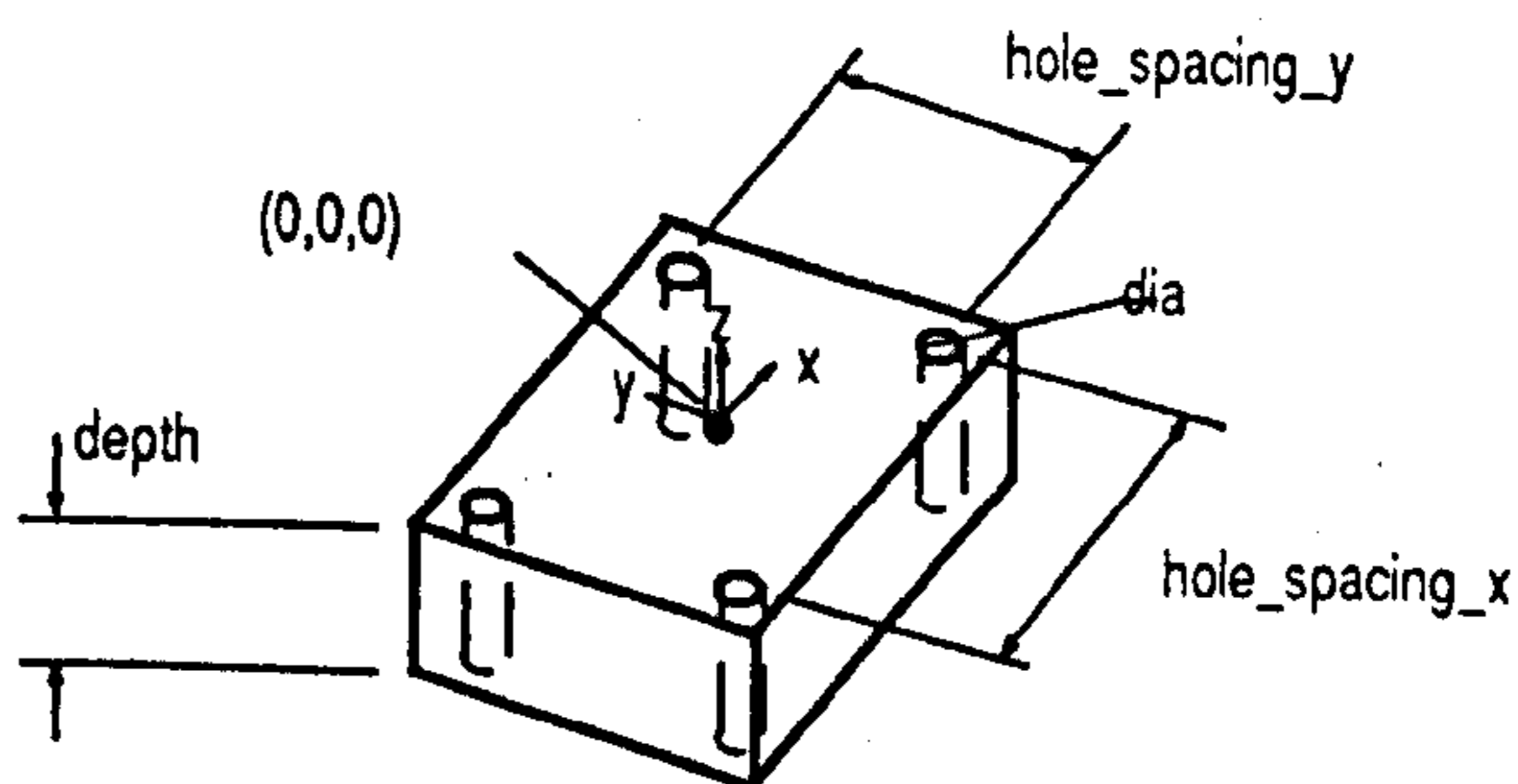
"thru hole with xy origin at centre, z at top

#### 4.4.7 The Four Corner Hole

```

four_corner_hole == (
  dia == ?? ,
  depth == ?? ,
  surf fin == ?? ,
  xdatum == ?? ,
  ydatum == 0.00000E+00,
  zdatum == ?? ,

```





```

hole_spacing_x == ?? ,
hole_spacing_y == ?? ) =
thru hole 1 : thru hole with xy origin at centre:
  dia= dia
  depth= depth
  surface roughness: surf fin
  ref.posn:
  x: xdatum
  y: ydatum
  z: zdatum
  AT
  mx =- ( hole_spacing_x / 2.0 ),
      my =- ( hole_spacing_y / 2.0 )
thru hole 2 : thru hole with xy origin at centre:
  dia= dia
  depth= depth
  surface roughness: surf fin
  ref.posn:
  x: xdatum
  y: ydatum
  z: zdatum
  AT
  mx =- ( hole_spacing_x / 2.0 ),
      my = hole_spacing_y / 2.0
thru hole 3 : thru hole with xy origin at centre:
  dia= dia
  depth= depth
  surface roughness: surf fin
  ref.posn:
  x: xdatum
  y: ydatum
  z: zdatum
  AT
  mx = hole_spacing_x / 2.0,
      my = hole_spacing_y / 2.0
thru hole 4 : thru hole with xy origin at centre:
  dia= dia
  depth= depth
  surface roughness: surf fin
  ref.posn:
  x: xdatum
  y: ydatum
  z: zdatum
  AT
  mx = hole_spacing_x / 2.0,
      my =- ( hole_spacing_y / 2.0 )

```

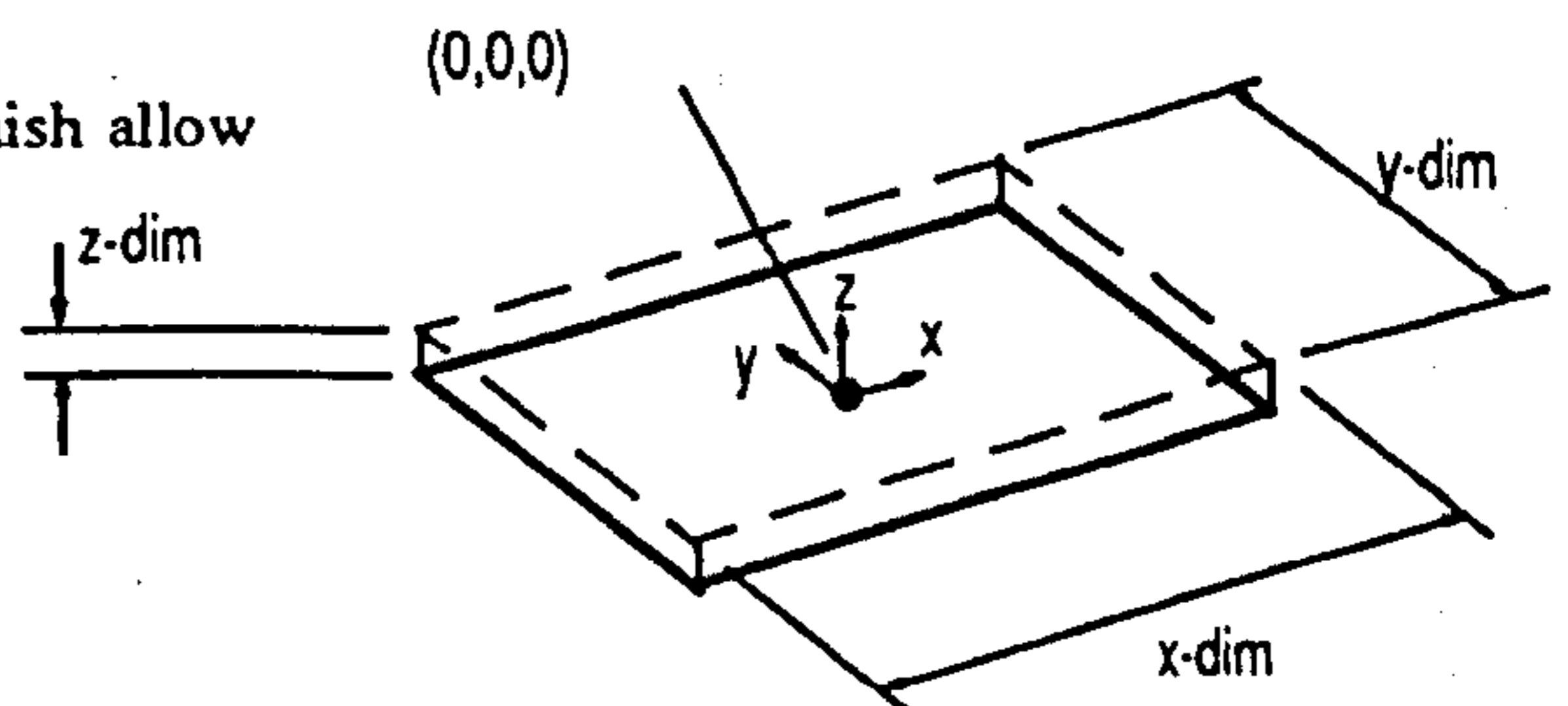
"4 hole set with xy origin at centre line intersection

4.4.8 The Face

```

moved face == (
  x-dim == body length,
  y-dim == body width,
  z-dim == ( stock height -
    body thickness + cleanup allow + finish allow
  ),
  surface roughnes == 2.5,
  xdatum == 0.00000E+00,
  ydatum == 0.00000E+00,
  zdatum == 0.00000E+00) =
Intersection:

```



```

+x face : Transformed solid:
  Plane
  AT
  ry = 90.0,mx = x-dim / 2.0
-x face : Transformed solid:
  Plane
  AT
  ry = -90.0,mx =- ( x-dim / 2.0 )
+y face : Transformed solid:
  Plane
  AT
  rx = -90.0,my = y-dim / 2.0
-y face : Transformed solid:
  Plane
  AT
  rx = 90.0,my =- ( y-dim / 2.0 )
+z face : Transformed solid:
  Plane
  AT
  mz = z-dim
-z face : Transformed solid:
  Plane
  AT
  rx = 180.0

```

## Intersection:

```

+x face : Transformed solid:
  Plane
  AT
  ry = 90.0,mx = x-dim / 2.0
-x face : Transformed solid:
  Plane
  AT
  ry = -90.0,mx =- ( x-dim / 2.0 )
+y face : Transformed solid:
  Plane
  AT
  rx = -90.0,my = y-dim / 2.0
-y face : Transformed solid:
  Plane
  AT
  rx = 90.0,my =- ( y-dim / 2.0 )
+z face : Transformed solid:
  Plane
  AT
  mz = z-dim
-z face : Transformed solid:
  Plane
  AT
  rx = 180.0

```

## machining data

```

reg_constraints:
  spindle_axes:
    minus_z
    all_xy
  tool_app_dirs:
    minus_z
    all_xy
  fix_face_norms:
    plus_z
region volume:
  x-dim * y-dim * z-dim

```

region length: x-dim  
 region width: y-dim  
 region depth: z-dim  
 operations:

machining method pre\_defined  
 basic op data  
   current cut roughing  
   approach dim ??  
   machining element  
     element number 5.0  
     element name clear\_face  
   constraints:  
     operation tolerance  
       ??  
     surface roughness  
       ??  
     diameter constraints  
       ??  
     radius constraints  
       ??  
     length constraints  
       ??  
 op type data  
   current op facing  
   width y-dim  
   depth ( z-dim - finish allow)  
   length x-dim  
   radius ??  
   stock width y-dim  
   width2 ??  
   depth2 ??

machining method pre\_defined  
 basic op data  
   current cut finishing  
   approach dim ??  
   machining element  
     element number 5.0  
     element name finish\_face  
   constraints:  
     operation tolerance  
       z-dim  
     surface roughness  
       ??  
     diameter constraints  
       ??  
     radius constraints  
       ??  
     length constraints  
       ??  
 op type data  
   current op facing  
   width y-dim  
   depth finish allow  
   length x-dim  
   radius ??  
   stock width y-dim  
   width2 ??  
   depth2 ??

"face block with origin at centre of bottom face



## 4.4.9 General Parameters

cleanup allow == 3.0  
 finish allow == 0.25

body length == [ 140.0, ( 0.1, -0.1)]  
 body width == [ 75.0, ( 0.2, -0.3)]  
 body thickness == [ 15.0, ( 0.2, -0.1)]

stock height == 47.0  
 stock\_width == [ 110.0, ( 1.0, 1.0)]  
 stock\_length == [ 150.0, ( 1.0, 1.0)]

min\_stock ==

cleanup allow + finish allow  
 rem\_stock\_length == ( stock\_length - body length + min\_stock)  
 rem\_stock\_width == ( stock\_width - body width + min\_stock)  
 rem\_stock\_height == ( stock height - body thickness + min\_stock)

## 4.5 COMPONENT DESCRIPTIONS

## 4.5.1 The Bolster Plate - Variant 2

1 Transformed solid:  
 2 block with origin at centre  
 3 x = body length  
 4 y = body width  
 5 z = body thickness AT  
 6  
 7 mx =( body\_length / 2.00 ),my =( body\_width / 2.00 ),  
 mz =- (body thickness / 2.00 ) +  
 8  
 9 left top pocket : Pocket with axes and top surf at (0,0,0)  
 10 x\_dim= 50.00  
 11 y\_dim= 40.00  
 12 z\_dim= 10.00  
 13 corner rad= 15.00  
 14 fillet rad= 1.00  
 15 surf fin (Cla microns): 2.50  
 16 ref posn:  
 17 x: 0.00  
 18 y: 0.00  
 19 z: 0.00 AT  
 20  
 21 rz = 90.00,mx = 110.00,my = 75.00  
 22 right top pocket : Pocket with axes and top surf at (0,0,0)  
 23 x\_dim= 50.00  
 24 y\_dim= 40.00  
 25 z\_dim= 10.00  
 26 corner rad= 15.00  
 27 fillet rad= 1.00  
 28 surf fin (Cla microns): 2.50  
 29 ref posn:  
 30 x: 0.00  
 31 y: 0.00  
 32 z: 0.00 AT  
 33  
 34 rz = 90.00,mx = 190.00,my = 75.00  
 35 left thru pocket :  
 Thru pocket defined with axes and top surface at (0,0,0)

```

36     Pocket dimensions:
37     length = 40.00
38     width = 30.00
39     depth = ({5 body thickness} - {12})
40     Corner radius = 10.00
41     Surf.fin (Cla microns): 2.50
42     Ref. Coords:
43     x: 0.00
44     y: 0.00
45     z: 0.00
46     AT
47
48     {21},{21},{21},mz =- ({12} )
49 right thru pocke :
      Thru pocket defined with axes and top surface at (0,0,0)
50     Pocket dimensions:
51     length = 40.00
52     width = 30.00
53     depth = ({5 body thickness} - 10.00)
54     Corner radius = 10.00
55     Surf.fin (Cla microns): 2.50
56     Ref. Coords:
57     x: 0.00
58     y: 0.00
59     z: 0.00
60     AT
61
62     rz = 90.00,mx = 190.00,my = 75.00,mz =- ( 10.00 )
63 left channel : channel length in x, (0,0,0) at mid x,y, top z
64     width = 35.00
65     length = {4 body width}
66     depth = 10.00
67     finish (Cla microns): 2.50
68     datum coords:
69     x = 0.00
70     y = 0.00
71     z = 0.00
72     AT
73
74     rz = 90.00,mx = 50.00,my ={4 body width} / 2.00
75 right channel : channel length in x, (0,0,0) at mid x,y, top z
76     width = 35.00
77     length = {4 body width}
78     depth = 10.00
79     finish (Cla microns): 2.50
80     datum coords:
81     x = 0.00
82     y = 0.00
83     z = 0.00
84     AT
85
86     rz = 90.00,mx = 250.00,my ={4 body width} / 2.00
87 front step : step length in x,(0,0,0) at front,left,top
88     width = 10.00
89     length = {3 body length}
90     depth = 15.00
91     finish (Cla microns): 2.50
92     fillet rad: 1.00
93     Ref posn:
94     x: 0.00
95     y: 0.00
96     z: 0.00

```

## 4.5.2 The Bolster Plate - Variant 4

part:

bolster : Geometry:Transformed solid:

block with origin at centre

x = body length

y = body width

z = body thickness AT

mx = body length / 2.0,

my = body width / 2.0,

mz =- ( body thickness / 2.0 ) +

left top pocket :

Pocket with axes and top surf at (0,0,0)

x\_dim= 25.0

y\_dim= 20.0

z\_dim= 5.0

corner rad= 7.0

fillet rad= 1.0

surf fin (Cla microns): 2.5

ref posn:

x: 0.00000E+00

y: 0.00000E+00

z: 0.00000E+00 AT

rz = 90.0,mx = 50.0,my = 37.5

right top pocket :

Pocket with axes and top surf at (0,0,0)

x\_dim= 25.0

y\_dim= 20.0

z\_dim= 5.0

corner rad= 7.0

fillet rad= 1.0

surf fin (Cla microns): 2.5

ref posn:

x: 0.00000E+00

y: 0.00000E+00

z: 0.00000E+00 AT

rz = 90.0,mx = 90.0,my = 37.5

left thru pocket :

Thru pocket defined with axes and top surface at (0,0,0)

Pocket dimensions:

length = 20.0

width = 15.0

depth = ( body thickness - 5.0)

Corner radius = 5.0

Surf.fin (Cla microns): 2.5

Ref. Coords:

x: 0.00000E+00

y: 0.00000E+00

z: 0.00000E+00

AT

rz = 90.0,mx = 50.0,my = 37.5,mz =- ( 5.0 )

right thru pocke :

Thru pocket defined with axes and top surface at (0,0,0)

Pocket dimensions:

length = 20.0

width = 15.0

depth = ( body thickness - 5.0)

Corner radius = 5.0

Surf.fin (Cla microns): 2.5



```

Ref. Coords:
x: 0.00000E+00
y: 0.00000E+00
z: 0.00000E+00
  AT
  rz = 90.0,mx = 90.0,my = 37.5,mz =- ( 5.0 )
front step :
step length in x,(0,0,0) at front,left,top
width = 5.0
length = body length
depth = 7.0
finish (Cla microns): 2.5
fillet rad: 1.0
Ref posn:
x: 0.00000E+00
y: 0.00000E+00
z: 0.00000E+00
  thru holes_12mm :
  4 hole set with xy origin at centre line intersection
dia of holes: 12.0
depth: body thickness
surface finish: 2.0
ref.pasn:
x: 0.00000E+00
y: 0.00000E+00
z: 0.00000E+00
hole spacing:
x= 108.0
y= 35.0
  AT
mx = body length / 2.0,
      my = body width / 2.0
top face :
face block with origin at centre of bottom face
x-dim= stock_length
y-dim= stock_width
z-dim= rem_stock_height
surf fin (Cla microns): 2.5
ref point:
x: 0.00000E+00
y: 0.00000E+00
z: 0.00000E+00
  AT
      mx =( stock_length / 2.0 - rem_stock_length),
      my =( stock_width / 2.0 - min_stock)
bottom face :
face block with origin at centre of bottom face
x-dim= stock_length
y-dim= stock_width
z-dim= min_stock
surf fin (Cla microns): 2.5
ref point:
x: 0.00000E+00
y: 0.00000E+00
z: 0.00000E+00
  AT
      rx = 180.0,
      mx =( stock_length / 2.0 - rem_stock_length),
      my =( stock_width / 2.0 - min_stock),
      mz =- ( body thickness )
back face :
face block with origin at centre of bottom face
x-dim= stock_length

```

```

y-dim= stock height
z-dim= rem_stock_width
surf fin (Cla microns): 2.5
ref point:
x: 0.00000E+00
y: 0.00000E+00
z: 0.00000E+00
AT
    rx =- ( 90.0 ),
    mx =( stock_length / 2.0 - rem_stock_length),
    mz =- (( stock height / 2.0 - rem_stock_height)),
    my = body width
front face :
face block with origin at centre of bottom face
x-dim= stock_length
y-dim= stock height
z-dim= min_stock
surf fin (Cla microns): 2.5
ref point:
x: 0.00000E+00
y: 0.00000E+00
z: 0.00000E+00
AT
    rx = 90.0,
    mx =( stock_length / 2.0 - rem_stock_length),
    mz =- (( stock height / 2.0 - rem_stock_height) )
left face :
face block with origin at centre of bottom face
x-dim= stock height
y-dim= stock_width
z-dim= rem_stock_length
surf fin (Cla microns): 2.5
ref point:
x: 0.00000E+00
y: 0.00000E+00
z: 0.00000E+00
AT
    ry =- ( 90.0 ),
    my =( stock_width / 2.0 - min_stock),
    mz =- (( stock height / 2.0 - rem_stock_height) )
right face :
face block with origin at centre of bottom face
x-dim= stock height
y-dim= stock_width
z-dim= min_stock
surf fin (Cla microns): 2.5
ref point:
x: 0.00000E+00
y: 0.00000E+00
z: 0.00000E+00
AT
    ry = 90.0,
    my =( stock_width / 2.0 - min_stock),
    mz =- (( stock height / 2.0 - rem_stock_height) ),
    mx = body length

```

### 4.5.3 The Glacier Reduced Size Bearing

solid definition :  
description :  
E3 geometry :

base plate geometry :

SOLID:

shape definition : geometry application:

geometry : machined block

( finished x : overall length ,  
pre-machined x : stock length ,  
finished y : overall depth ,  
pre-machined y : stock depth ,  
finished z : height ,  
pre-machined z : stock height ,  
finish cut : finishing allowance ,  
rough cut : roughing allowance ,  
surface finish : surface finish )

applied geometry : the features : as ordered

( moved slot : transformed applied geometry  
( rigid motions : ( mz ( half ( height ) ) )  
applied geometry : slot : channel  
( width : slot width ,  
depth : slot depth ,  
measured width : 13.5 ,  
measured depth : 5.5 ,  
length : overall length ,  
finish cut : finishing allowance ,  
surface finish : surface finish ) )

through holes : as ordered :

( +x holes : as ordered :

( +x+y hole : transformed applied geometry  
( rigid motions : ( mx ( half x centres : ( half \* through hole x centres ) )  
my ( half y centres : ( half \* through hole y centres ) ) ) )  
applied geometry : through hole : through hole  
( diameter : through hole diameter ,  
measured diameter : 10.0 ,  
depth : height ,  
surface finish : surface finish ) ) )

+x-y hole : transformed applied geometry

( rigid motions : ( mx ( minus x centres : ( - through hole x centres ) ) )  
applied geometry : +x+y hole : transformed applied geometry  
( rigid motions : ( mx ( half x centres : ( half \* through hole x centres ) )  
my ( half y centres : ( half \* through hole y centres ) ) ) )  
applied geometry : through hole : through hole  
( diameter : through hole diameter ,  
measured diameter : 10.0 ,  
depth : height ,  
surface finish : surface finish ) ) ) )

-y holes : transformed applied geometry

( rigid motions : ( my ( minus y centres : ( - through hole y centres ) ) )  
applied geometry : +x holes : as ordered :

( -x-y hole : transformed applied geometry

( rigid motions : ( mx ( half x centres : ( half \* through hole x centres ) )  
my ( half y centres : ( half \* through hole y centres ) ) ) )  
applied geometry : through hole : through hole  
( diameter : through hole diameter ,  
measured diameter : 10.0 ,  
depth : height ,  
surface finish : surface finish ) )



```

+x-y hole : transformed applied geometry
  ( rigid motions : ( mx (
    minus x centres : ( - through hole x centres ) ) )
    applied geometry : +x+y hole : transformed applied geometry
  ( rigid motions : ( mx (
    half x centres : ( half * through hole x centres ) )
    my (
    half y centres : ( half * through hole y centres ) ) )
    applied geometry : through hole : through hole
    ( diameter : through hole diameter ,
      measured diameter : 10.5 ,
      depth : height ,
      surface finish : surface finish ) ) ) ) ) )
material : Grade 17 cast iron
evaluated geometry :
relationship graph : coordinate relationship:
planned processes : sequential set :
decision network :
-----

```

## APPENDIX 5

## ELEMENTS OF THE IMPLEMENTED MACHINE PLANNER CODE

## 5.1 LINKS TO THE PRODUCT DATA MODEL

The following listing of code illustrates the link from the PDM to the Machine Planning software, enabling the system user to draw data from a Product Model into the the Machine Planner, to use the Machine Planning functions, and to put data back into the Product Model.

```

if choice.variant = 1 -- Return
then
    main_choice := 0; -- return to main menu

elsif choice.variant = 2 -- Attach Tooling Data Base
then
    node_val := tooling_db_cds.walk2(i);
    convert;

elsif choice.variant = 3 -- get machine data
then
    float_val := machine_tool_cds.walk2(i);

elsif choice.variant = 4 -- Interactive dialogue for user data
then
    background.cds_user_option
        (select_opt, tl_opt, input_tool_life);
    --
    setup := new setup_rec;
    part.setups := new setup_rec;
    part.setups := setup;
    background.get_environment (part);
    setup := part.setups;

elsif choice.variant = 5 -- get component data
then
    ii := lambda_evaluation.expand ( i );
    walk_res := sei_pp_comp_defn.walk2 ( ii.sons.tl.hd );
    screen_text_io.put ("end walking");
    material_info.material := pp_mp_conversions.convert ( walk_res );
    part := pp_mp_conversions.convert ( walk_res );
    part.geom := pp_mp_conversions.convert ( walk_res );
    region := pp_mp_conversions.convert ( walk_res );
    manip_setupsnregions.goto_first_reg_in_list (region);
    part.regions := region;

elsif choice.variant = 6 -- create sds
then
    --- get data from component level
    display.current_inst := lambda_evaluation.expand ( i );
    sei_gg_e3_rigid_motions_makes.current_transform
        := three_transforms.identity ;
    screen_text_io.new_line;

```

```

if spelling(display.current_inst.form.node_id).all = "E3 solid" then
screen_text_io.put ( "Creating CSG tree" );
tree := sei_gg_solid.walk2 (display.current_inst);
else
screen_text_io.put ( "Wrong node type!" & ascii.bel );
end if ;

--
screen_text_io.new_line;
screen_text_io.put ( "calculating bounds" );
----- lower bounds first then upper
bounds := sdsd_pack_ry.get_bounds(
(- part.geom.length.num_value / 2.0) +
part.geom.posn.movex.num_value ,
(- part.geom.width.num_value / 2.0) +
part.geom.posn.movey.num_value ,
(- part.geom.depth.num_value / 2.0) +
part.geom.posn.movez.num_value ,
( part.geom.length.num_value / 2.0) +
part.geom.posn.movex.num_value ,
( part.geom.width.num_value / 2.0) +
part.geom.posn.movey.num_value ,
( part.geom.depth.num_value / 2.0) +
part.geom.posn.movez.num_value );
tree.bounds:= bounds;
-----
screen_text_io.new_line;
screen_text_io.put ( "producing sdsd" );
component := sdsd_pack_ry.produce_sdsd (tree, bounds);
part.geom.sdsd_object := component;
reg := part.regions;
while reg /= null
loop
screen_text_io.new_line;
screen_text_io.put ("finding region cell address");
reg.geom.address := sdsd_manip.get_region_cell_address (part, reg);
reg := reg.next;
end loop;

elsif choice.variant = 7 -- id setups
then
comp_setups_sequ_method (part, region, setup);

elsif choice.variant = 8 -- identify operations list for setup
-- and select possible tools
then
first_setup := setup;
while setup /= null
loop
manip_setupsnregions.extract_ops_from_regions_list
(setup, setup.regions, operation);
head := operation;
while operation.succ /= null
loop
operation := operation.succ;
tool_select (operation, part,tl_opt, input_tool_life, s_editor);
end loop;
setup.operations := head.succ;
setup := setup.next;
end loop;
setup := first_setup;
part.setups := setup;

elsif choice.variant = 9 -- sequ. ops and identify best tools

```



```

then
    setup := part.setups;
    while setup /= null -- sequence ops in each setup
    loop
        setup.machine_no := 1; -- this needs to elsewhere, op has the mc no.
        sequence_ops ( part, setup, tl_opt, input_tool_life );
        setup_info (part, setup);
        put_setup_info (setup);
        setup := setup.next;
    end loop;
    setup := first_setup;
    -- file planning and detail setup data for each comp. setup
    file_setup_info (part, setup);

elsif choice.variant = 10 -- identify m/c batch info
then
    identify_fix_datum_posns (part);

elsif choice.variant = 11 -- extract NC code to file.
then
    while setup /= null -- extract part progs for each setup
    loop
        nc_extraction.extract_nc_code (part, setup );
        setup := setup.next;
    end loop;
    setup := first_setup;

elsif choice.variant = 12 -- produce drawing
then
    graphics . init_graf ;
    sdsqm_query . draw_sdsqm_boundary ( component . root_cell ) ;

elsif choice.variant = 13 -- remove drawing
then
    graphics . end_graf ;

elsif choice.variant = 14 -- dims and tols
then
    --- NOTE: this has a bug in "Evaluating Relationship Graph
    screen_text_io.put ( "Reading Component data" );
    the_comp_def_i := i.sons.tl.hd ;
    the_comp_def := sei_rge_comp_defn.walk2 (the_comp_def_i) ;
    the_e3_geom := sei_mgg.e3_geometry_pack.walk2
        ( the_comp_def.the_single_material.the_e3_geometry ) ;

    screen_text_io.put ( "Evaluating Relationship Graph" );
    text_io.new_line;
    the_rel_graph := rge_geom_walk.rg_from_e3_geom_defn
        (the_e3_geom.the_definition) ;

    text_io.put ( "Walking Relationship Graph" );
    rg_walking_ry.mp_data_analysis(the_rel_graph.the_x_node ) ;
    rg_walking_ry.mp_data_analysis(the_rel_graph.the_y_node ) ;
    rg_walking_ry.mp_data_analysis( the_rel_graph.the_z_node ) ;

elsif choice.variant = 15 -- put back setup data
then
    se_sequ_setup := mp_pp_conversions.convert(setup);
    screen_text_io.put (" put back setup data");
    screen_text_io.put ("create node inst");
    if sundries.spelling (display.current_inst.form.node_id).all =
    "setup activity"
    then

```

```

sei_pp_setup_activity.create (se_sequ_setup, display.current_inst);
  else
    screen_text_io.put_line ( "Node should be a {setup activity}" );
  end if ;
else
  screen_text_io.put("No action taken");
end if;

```

## 5.2 THE SETUPS SEQUENCE METHOD

Option 7 in software listing provided in section 5.1 calls the comp\_setups\_sequ\_method. The following listing illustrates the main procedures used in this procedure which produces a setup sequence.

---- procedures which identify features data beyond that  
 ---- which is identified directly from the PDM interface

```

identify_reg_types          (part, regions);
ident_vol_removal_regs      (part, regions);
identify_reg_posns_ona_component (part, regions);
-----

--- group regions by spindle axis ---
group_regions_wrt_appdim    (part, regions, setups );
-----

--- identify geometric relationships
setup := setups;
while setup /= null
loop
update_reg_posns_by_setup   (part, setup);
setup := setup.next;
end loop;
-----

--- apply setup rules ---
part.constraint.min_stock_dim_ratio :=
prefered_min_stock_dim_ratio_rule (part);
part.constraint.crit_tol_grouping :=
crit_tol_for_grouping_rule (part, machines.machine);
part.constraint.vol_ratio_bulk_removal :=
early_bulk_removal_vol_ratio_rule (part );
group_setups_by_no_regions (setups);
ident_vol_removal_setups (part, setups );
group_setups_by_bulk_removal (part, setups );
group_setups_wrt_crit_tols (part, setups );
identify_likely_fix_regions (part, setups );
prin_setup := setups;
identify_rel_setup_posns (part, prin_setup, top, bottom, front,
back, left_side, right_side );
-----

--- select fixturing strategy ---
top.constraint.side_clamping_allowed :=
side_clamping_allowed_rule (part, top);
  case top.constraint.side_clamping_allowed is
  when true =>
    side_clamp_method(part, setups, top, bottom, back,
front, left_side, right_side, first_setup);
  when false =>
    if prin_setup_is_thru_clamped (top)

```

```

        then
        thru_clamping_method (part, top, bottom, back, front, left_side,
        right_side, first_setup);
        else
        down_clamping_method (part, top, bottom, back, front, left_side,
        right_side, first_setup) ;
        end if;
first_setup := setups;
-----

-- update setup data ---
remove_dup_regions_from_setuplist (setups);
remove_unnec_setups_from_setuplist (setups);
part.setups := setups;
-----

```

### 5.3 THE THROUGH CLAMPING STRATEGY

The following listing illustrates the "through\_clamping\_method", showing that the principal setup is checked for suitable holes. Assuming these are found, preliminary setups are identified, datum and clamping is finalised, and the side Face features are converted to "profiles".

```

-- find possible through clamp regions for principal setup
---
goto_first_reg_in_list (top.regions);
while top.regions /= null
loop
if top.regions.suit_for_fix.clamp = ideal
and top.regions.kind = thru_hole
then
if clamp_regs = null
then
clamp_regs := new region_rec;
clamp_regs.all := top.regions.all;
clamp_regs.next := null;
clamp_regs.last := null;
else
new_reg := new region_rec;
new_reg.all := top.regions.all;
new_reg.last := clamp_regs;
new_reg.next := null;
clamp_regs.next := new_reg;
clamp_regs := new_reg;
end if;
end if;
exit when top.regions.next = null;
top.regions := top.regions.next;
end loop;
-----

---check available through holes have clearance
---
reg_list := clamp_regs;
---
id_regs_with_bolt_clearance (part, top, clamp_regs, top.regions);
--- uses "clear_around_region" to check clearance
---
count_regs (clamp_regs, count);
case count is

```



```

when 0 =>
-- this leads to down clamping
down_clamping_method (part, top, bottom, back,
    front, left_side, right_side, first_setup);
when 1 =>
down_clamping_method (part, top, bottom, back,
    front, left_side, right_side, first_setup);
-- this leads to hole + blind hole method
-- down clamp used for now
when 2 =>
-- this is ok for thru method, but datum will be on fixture
-- or more prior machining must be done to give one.
-- so down clamp used for now
down_clamping_method (part, top, bottom, back,
    front, left_side, right_side, first_setup);
when others =>
    id_regs_with_greatest_spacing (clamp_regs, clamp1, clamp2, spacing);
    id_datum_hole (reg_list, clamp1, clamp2, datum_reg);
---
---
---
    create_prelim_setup (top, new_setup, hold_reg);
--
--
    find_datum_and_detail_clamping_for_prin_setup (part, top);
--
--
    change_face_to_profile (back, top);
    change_face_to_profile (right_side, top);
    change_face_to_profile (front, top);
    change_face_to_profile (left_side, top);
--

```

#### 5.4 THE CLEAR AROUND FEATURE QUERY

The following listing illustrates the "clear\_around\_region\_query" which starts in a feature, moves to the edge of a feature and then checks for feature interactions in each of the perpendicular directions, relative to the spindle axis,

```

case spindle_axis is
when z =>
edge_px := next_valid_cell
    (rep      => rep,
     in_cell  => mid_cell,
     axis     => x,
     pos      => true );
clear_in_dim := clear_dist
    (rep      => rep,
     in_cell  => edge_px,
     dist     => requ_dist,
     axis     => x,
     pos      => true);
if clear_in_dim = false
then return false;
end if;

edge_nx := next_valid_cell
    (rep      => rep,
     in_cell  => mid_cell,
     axis     => x,

```

```

        pos      => false );
clear_in_dim := clear_dist
    (rep      => rep,
     in_cell  => edge_nx,
     dist     => requ_dist,
     axis     => x,
     pos      => false);
if clear_in_dim = false
then return false;
end if;

edge_py := next_valid_cell
    (rep      => rep,
     in_cell  => mid_cell,
     axis     => y,
     pos      => true );
clear_in_dim := clear_dist
    (rep      => rep,
     in_cell  => edge_py,
     dist     => requ_dist,
     axis     => y,
     pos      => true);
if clear_in_dim = false
then return false;
end if;

edge_ny := next_valid_cell
    (rep      => rep,
     in_cell  => mid_cell,
     axis     => y,
     pos      => false );

clear_in_dim := clear_dist
    (rep      => rep,
     in_cell  => edge_ny,
     dist     => requ_dist,
     axis     => y,
     pos      => false);
if clear_in_dim = false
then return false;
end if;
---
---
--- similarly for other spindle axis directions
---
---
end case;
return true;

```

### 5.5 THE IDENTIFY CUTTING TOOLS PROCEDURE

The following listing illustrates the identify cutting tools procedure, which finds possible tools from a list extracted from a database.

```

-- call machining rules
machining_rules.tooling_rules (part, operation);
--
--
for I in 1..NO_OF_CUTTERS loop
if CUTTER(I).CUTTER_TYPE = OPERATION.REQ_CUTTER.TYPE1
or CUTTER(I).CUTTER_TYPE = OPERATION.REQ_CUTTER.TYPE2

```

```

then
if CUTTER(I).CUTTER_DIA >= OPERATION.MIN_CUTTER_DIA
and CUTTER(I).CUTTER_DIA <= OPERATION.MAX_CUTTER_DIA
then
if cutter(I).length1 >= operation.min_cutter_len
then
identify_tip_data (operation, I, part, tl_opt, input_tool_life);
end if;
end if ;
end if;
end loop ;

```

## 5.6 THE CALCULATION OF PROCESSING DATA

The following listing provides key procedures in the calculate processing data procedure. Cutting rules are shown as being called from this procedure. In future implementations they should be repositioned at the end of the tool selection procedure. This will enable links to general feature evaluation to be made.

```

machining_rules.cutting_rules (Operation, J);
calc_cut_depth (Operation, J);
calc_cut_width (Operation, J);
nc_extraction.find_path_lengths (operation); -- calculate path lengths
case operation.current_op is
when drilling | reaming | tapping =>
find_hole_cutdata (operation, part);
when facing | end_milling | open_slotting | closed_slotting | composite =>
feedtooth (Operation, J);
calc_tool_life (Operation, J, tl_opt, input_ttol_life, part);
--
loop
--
repeat := false;
calc_speednfeed (Part, Operation, J);
--
if Operation.poss cutter.val.req_power > Operation.poss cutter.val.mc_power
then repeat := true;
end if;
if repeat = true
then
--
Operation.poss cutter.val.detail.no_depth_passes :=
Operation.poss cutter.val.detail.no_depth_passes + 1;
Operation.poss cutter.val.detail.cut_depth :=
Operation.op_depth / gmp.real (Operation.poss cutter.val.detail.no_depth_passes);
nc_extraction.find_path_lengths (operation); -- update path lengths
end if ;
exit when repeat = false;
end loop;

```

## 5.7 THE SEQUENCE OPERATIONS PROCEDURE

The following listing provides key procedures in the sequence operations procedure, highlighting in particular the "sequence\_using\_tools" procedure, which optimises the selection of tooling on the basis of path time.



```

seq_using_cut_order (setup, cut_seq );
seq_using_op_order (setup, cut_seq, op_seq);
seq_using_constraint_dias ( op_seq );
setup.operations := new operation_data;
setup.operations := op_seq;
-- sequence_using_tools (part, setup, tl_opt, input_tool_life);

operation := setup.operations;
while operation.succ /= null    -- to last op
loop
operation := operation.succ;
end loop;
--- backtrack through list finding best cutting data
while operation /= null
loop
if operation.current_op /= unknown
then
init_posstool := operation.posscutter;
while operation.posscutter /= null
loop
find_best_data (part, operation, j ) ;
operation.posscutter := operation.posscutter.next;
end loop;
operation.posscutter := init_posstool;
end if;
operation := operation.pred;
end loop;
-- now go forward thru list checking the best data
operation := setup.operations;
while operation /= null
loop
if operation.current_op /= unknown
then
check_best_data (operation);
put_best_data (operation);
check_data ( part, operation, tl_opt, input_tool_life);
end if;
end if;
operation := operation.succ;
end loop;

```

## 5.8 THE EXTRACTION OF MACHINE CONTROL CODE

The following listing provides sections of the extract machine control code procedure, showing how the initial code for a part program is produced, followed by the code for each machining Operation and finally by the code to finish the part program.

```

p_prog_name.name.all := part.no.name.all & "_setno" & setup_noval & ".prog";
p_prog.length := p_prog_name.length;
p_prog.name (1..p_prog.length) := p_prog_name.name.all;

--

operation := setup.operations;
while operation /= null
loop
if operation.pred = null
then
new_code (p_prog);
make_setup.start (
p_prog.tail_code.name,      -- file name

```

```

p_prog_name.name.all (1..6 )); -- name in p_prog
new_code (p_prog);
tool_change_req := true; -- this is true because its the first op.
call_requd_datum_set (p_prog,
p_prog.tail_code.name , -- file name
setup, -- data from setup
tool_change_req );-- boolean to identify if tool
-- is already in spindle
end if;

if operation.current_op = unknown
then null;
else
tool_change_req := true;
set_offsets (operation, x_offset, y_offset, z_offset);
op_datum_to_safe_rap :=
setup.safe_rapid_ht - z_offset;
-- dist from current op datum to safe rapid ht.
t_code := operation.best_tool.detail.cutter_no;
if operation.pred = null
then
tool_change_req := true;
elsif
operation.pred.best_tool.detail.cutter_no =
operation.best_tool.detail.cutter_no
then
tool_change_req := false;
elsif
operation.pred.current_op = unknown
then
last_op := operation.pred;
while last_op.current_op = unknown
loop
last_op := last_op.pred;
end loop;
if
last_op.best_tool.detail.cutter_no =
operation.best_tool.detail.cutter_no
then
tool_change_req := false;
end if;
else
tool_change_req := true;
end if;
init_poss_tool := operation.poss_cutter ;
loop
exit when
operation.best_tool.detail.cutter_no =
operation.poss_cutter.val.detail.cutter_no;
operation.poss_cutter := operation.poss_cutter.next;
end loop;
new_code (p_prog);
case operation.cut_method is
when pre_defined =>
case operation.machining_el.no is
---
--- example of one option follows
---
when 2 => --- call profile pocket procedure
make_cutter_path.profile_pocket (
p_prog.tail_code.name,
operation.transform.rotz,
operation.machining_el.pro_name,

```

```

    t_code,
    setup.no,
    x_offset,
    y_offset,
    z_offset,
    operation.op_length,
    operation.op_width,
    operation.op_depth,
    operation.comer_rad,
    operation.stock_width,
    operation.posscutter.val.detail.cut_width,
    operation.posscutter.val.detail.no_width_passes,
    operation.posscutter.val.detail.cut_depth,
    operation.posscutter.val.detail.no_depth_passes,
    operation.posscutter.val.detail.cutter_dia,
    operation.posscutter.val.feed_rate,
    operation.posscutter.val.feed_rate,
    operation.posscutter.val.speed,
    tool_change_req,
    op_datum_to_safe_rap );
if operation.succ = null
then
new_code (p_prog);
make_setup.finish (p_prog.tail_code.name, t_code );
compile_code (p_prog);
nc_support.standardise_spaces (p_prog_name.name.all);
renumber.nc_code (p_prog_name.name.all);
end if;
exit when operation.succ = null;
operation := operation.succ;
end loop;

```

## 5.9 MACHINING RULES

This section describes the rules used in the experimental Machine Planner implementation. These rules have been used to enable the exploratory planner implementation to function and are not intended as an exhaustive set of machining rules.

### 5.9.1 Cutting Rules

Cutting rules provide a basis on which a number of questions can be answered. These questions are as follows:

- What is the maximum percentage of cutter width which should be used in machining different types of Operation?
- What is the maximum percentage of flute length which should be used in machining different types of Operation?
- What is the value of surface roughness which should be used when calculating feed rates?
- Should coolant be used?
- If so, what coolant should be used?

The first two questions identify the geometric constraints on the use of a particular cutter to machine a specific operation. The last two questions consider the material being machined and identify whether coolant is needed and, if so, which should be use. The surface roughness query compares the required surface roughness with the required tolerance, to identify the appropriate value of surface roughness to use, to achieve both requirements.



This follows the recommendations laid down in the Machinability Data handbook [95] on surface roughness and dimensional tolerance comparisons. The ADA functions used in the research to answer these queries were:

```
function CHECK_OP_RA_RULE (OPERATION : in LINK) return gmp.real;
function MAX_CUT_WIDTH_RULE (OPERATION : in LINK ) return gmp.real;
function MAX_CUT_DEPTH_RULE (OPERATION : in LINK) return gmp.real;
function NEED_COOLANT_RULE (PART : in part_data;
                           OPERATION : in link) return BOOLEAN;
function FIND_COOLANT_TYPE_RULE (PART : in part_data;
                                OPERATION : in link) return MY_NAME;
```

### 5.9.2 Tooling Rules

As is the case with cutting rules, questions which relate to the suitability of cutters for use in machining an Operation can be identified, and rules used to provide answers. The queries addressed were as follows:

- What cutter types are appropriate in machining a specific Operation?
- What type of tip should be used?
- What grade of tip material should be used?
- What is the maximum cutter diameter which should be considered?
- What is the minimum cutter diameter which should be considered?
- What is the minimum cutter length which should be considered?

The ADA functions implemented in the research to answer these queries were:

```
function MAX_CUTTER_DIA_RULE ( OPERATION : in LINK)
                           return MAX_CUT_DIA_TYP;
function MIN_CUTTER_DIA_RULE ( OPERATION : in LINK)
                           return MIN_CUT_DIA_TYP;
function OP_TIP_REF3_RULE ( OPERATION : in LINK)
                           return TIP_REF3_TYP;
function OP_TIP_GRADE_RULE ( PART : in PART_DATA;
                             OPERATION : in LINK)
                           return TIP_GRADE_TYP;
procedure FIND_REQ_CUTTER_TYPES_RULE (OPERATION : in LINK ;
                                       VAL_TYPE1 : out VAL_REQ_CUTTER_TYP ;
                                       VAL_TYPE2 : out VAL_REQ_CUTTER_TYP);
```

### 5.9.3 Setup Rules

The rules used in setup planning can be grouped as general setup rules and rules which identify the suitability of features for fixturing. The implemented rules were:

```
function preferred_min_stock_dim_ratio_rule (part : part_data)
                                           return gmp.real;
function crit_tol_for_grouping_rule (part : part_data;
                                     machine : machines.mc_type_data)
                                   return gmp.real;
function early_bulk_removal_vol_ratio_rule (part : part_data)
                                           return gmp.real;
function side_clamping_allowed_rule (part : part_data;
                                     setup : setup_rec_ptr)
                                   return boolean;
function region_suitability_for_prim_loc_rule (part : part_data;
                                               region : region_ptr)
                                             return suitability_typ;
function region_suitability_for_sec_loc_rule (part : part_data;
```

```
region : region_ptr)
return suitability_typ;
function region_suitability_for_clamp_rule (part : part_data;
region : region_ptr)
return suitability_typ;
```

#### 5.9.4 Operation Sequencing Rules

Operation sequencing rules are employed to identify sequencing constraints. The rules used identify both cut type sequence, i.e. that roughing operations are performed before finishing Operations, and Operation type sequence, i.e. the order of preference between milling and drilling Operations etc. The resulting implemented rules are:

```
function current_cut_order_rule (setup : in setup_params.setup_rec_ptr)
return setup_params.cut_order_ptr;
function current_op_order_rule (setup : in setup_params.setup_rec_ptr)
return setup_params.op_order_ptr;
```

## APPENDIX 6

### A USER GUIDE TO THE SOFTWARE

#### 6.1 INTRODUCTION

This appendix describes how the Closely Coupled Machine Planning software is used. It uses the Glacier Workpiece, described in the integrated experiment in chapter 11.4, as the basis for a walk through of the system. Structure Editor commands are illustrated in double quotes. File names are illustrated in single quotes. It is assumed that a "carriage return" is required after each command.

#### 6.2 USING THE MACHINE PLANNING SOFTWARE

##### 6.2.1 Files Required

Three files are needed to run the Machine Planning software. These are an executable file called 'main'; a Structure Editor file, which represents the PDM, called 'pms3\_8.lam'; and another Structure Editor file, which contains the Product Model representation of the Glacier Workpiece, called 'demo.pms3\_8'. It is assumed that these files are in the directory in which the program is to run.

##### 6.2.2 Entering the Program

To start the program type 'main'. This starts the SE main program which provides the user with a menu as follows:

Make selection from:

- 1 VT100
- 2 TVI920 or DT22
- 3 TVI925
- 4 VT52
- 5 Insight
- 6 Sun Controller
- 7 Any other type

Choose option 6 (it is assumed the software is being run on a SUN workstation). The system will reply with the prompt:

```
Sun Controller
Meta-Structure definition
Please supply file name
Type character string
Input a character string
```



At this prompt type in the name of the PDM file, i.e. 'pms3\_8.lam'. This file does not load instantly, so be patient. When it has loaded the system will return with the prompt:

```

Reading Data
Reading from /usr2/sun_b/iss/gmc/demo/pms3_8.lam
Feb 89 version
Meta-structure map complete
{ Corresponds to meta structure in DSKS4:[DAVEH.SED]MI1.MI1;1 }
Data reading complete
  Converting level0 definition to meta form
Transforming node
Feb 87 version
Starting to WALK
  Make_meta complete
11459 node_inst cells reclaimed.   15995 inst_list cells reclaimed
  Ready for file to be edited
Emptying name index
Make selection from:
  1 load from file
  2 interactive mode
  3 skeleton
  4 ditto nil lists

```

Select option 1 to run the demonstration software. The software will return the prompt:

```

load from file
Please supply file name
Type character string
Input a character string

```

At this point load the SE file containing the Glacier Workpiece Product Model, i.e. 'demo.pms3\_8'. The system will return:

```

Reading Data
Reading from /usr2/sun_b/iss/gmc/demo/demo.pms3_8
Meta-structure map complete
{ Corresponds to meta structure in /usr2/sun_b/iss/gmc/demo/pms3_8.lam }

```

This means the file is being loaded. This will take longer than the PDM file but the system will eventually return the following display to the screen:

```

{root} -- Collection = Alternate
PRODUCT RANGE:
...
Ready

```

This is the root description of the Product Model and can be traversed and used to drive the Machine Planning software as described in the following sections.

### 6.2.3 The Root Description of The Product Model

In the root description of the Product Model illustrated above, the "..." on the screen means that more data is contained in the structure, but it is hidden from view. The command "H" will provide the user with a deeper view of data. Using this command a few times will result in a screen image as:

{root} -- Collection = Alternate  
PRODUCT RANGE:

PRODUCT:

structural bearing :  
  data about data : ...  
  specification : ...  
  definitions : ...  
  actuals : ...

Ready

### 6.2.4 The "D" Option in the Structure Editor

The SE provides a "D" command which allows linked Applications to be called. When this is first used it will provide a screen image containing the following selections:

Make selection from:

- 1 Lambda evaluation
- 2 B-spline evaluator
- 3 SDSM evaluator
- 4 CDS interface
- 5 3D wireframe
- 6 Multiple evaluators
- 7 Dimensions & Tolerances
- 8 Machine planning
- 9 Inspection
- 10 Manf'g data analysis
- 11 Other options
- 12 Miscellaneous

Choose option 8. This will result in a further listing:

Machine planning

Make selection from:

- 1 Main menu
- 2 Features approach
- 3 General approach

Choose option 2. This will result in a further listing which provides the Author's Machine Planning Options:

Features approach

Make selection from:

- 1 Go to Main Menu
- 2 Tooling Database
- 3 Machine Data
- 4 User Data
- 5 Component Data
- 6 Create Sdsm
- 7 Plan setups
- 8 Find Poss.Tools
- 9 Sequence Ops
- 10 Find M/c Setups
- 11 Extract NC Code
- 12 Draw sdsd
- 13 Remove drawing
- 14 Dims and Tols
- 15 Put back data

Options 2 -> 6 should be called, in sequence, before any other options are used. Options 12 and 13 can then be used at any time. Options 7 -> 9 must be used in sequence, and must come before 10,11, and 15. Option 14 has not been implemented. Option 15 can be used before or after options 10 and 11. Before these Options can be used it is important to be at the correct point in the data structure in order to access the relevant data. The following sub-headings describe how each Option in the above listing can be used.

### 6.2.5 Go to Main Menu - Option 1

If the user is finished using the Machine Planning software, or wishes to move to an alternative Application, then this option should be used. This will take the user back up through the menus.

### 6.2.6 Tooling Database - Option 2

The tooling database data is contained under the {manufacturing information} node in the Product Model framework. This is reached by moving down and traversing to the right in the structure. Moving down is done by using the "d" command and moving to the right is done using the "r" command. "drrr" moves from the root to the {manufacturing information} node. "d" then moves down to the {tooling database} node. This is the position in the structure from which the tooling data can be loaded using option 2. When this has been done the SE will return "Ready" to the screen

### 6.2.7 Machine Data - Option 3

Data representing the Wadkin V4-6 is found to the right, and one level down from the {tooling database} node. "rd" moves to it. Option 3 can be used and again the SE will return "Ready" to the screen.

### 6.2.8 User Data - Option 4

This option is used to identify the basis on which tool life is to be calculated and can be used at any point in the structure. Calling this option leads to a further menu selection:

#### User Data

Please choose tool life option.

- 1 Maximise Production Rate.
- 2 Minimise Cost.
- 3 Manual Input of Tool Life.

Selection of options 1 or 2 will lead to the use of an appropriate equation from which the tool life will be calculated. Selection of option 3 will lead to the user being asked to provide a value for the tool life, in minutes, to be used:

Input the value of tool life (mins) to be used.

Input a real number

A real number, not an integer, must to be used.

### 6.2.9 Component Data - Option 5

This Option extracts the component data, needed by the Machine Planner, from the Product Model. The data which is extracted includes the features, which are defined within the component definition; the stock dimensions, bounding box and volume; the component di-



mensions, bounding box and volume; and the material specification.(refer to chapter 9.3.1). Before this option can be used it is important to be positioned at the correct node in the structure. This is achieved by using the following commands:

"U" - moves back to the root;  
 ""component" - sets the node {component} as a node to be searched for;  
 "N" - finds the {component} node;  
 "+zz" - shows the data structure at the {component} node;  
 "drdrd" - moves to the {definition} node which is used in this option

Having reached this point in the structure, option 5 can be called.

### 6.2.10 Create Sdsm - Option 6

This option links to the SDSM Application and generates an SDSM model of the component. This option is called from an {E3 solid} node. This is reached by using the following pair of commands: ""E3 solid" and "N". This will move down to the {E3 solid} node, under which lies the geometric representation of the component. Once at this node in the structure option 6 can be called which will produce the SDSM representation of the component. It also identifies the SDSM cell addresses for each of the feature reference coordinates. This then provides the basis for feature interaction checks which are contained in the next option.

### 6.2.11 Plan setups - Option 7

This option takes the data which has been extracted from the model in options 5 and 6 and performs part of the *identify features* activity described in chapter 9.3.1 and the *sequence setups* activity described in chapter 9.3.2. It results in a sequence of setups which have to be finalised using option 8. There is no need to be at any specific node when calling this option.

The software used when option 7 is called is listed in appendix 5.2. Before option 7 is called, features have already been extracted from the Product Model (option 5). The first procedures in option 7 complete the *identify features* requirement. Holes and Faces are identified as fixturing features (refer to chapter 9.3.1). The total volume of material to be removed from the stock is calculated and the region positions on the component are identified (refer to chapter 9.3.1).

The remaining procedures under option 7 fulfil the *sequence setups* requirement. These procedures perform the following functions

1. group the features according to their spindle axes; (refer to chapter 9.3.3.2.1)
2. identify the relative position of each feature, with respect to each setup;(refer to chapter 9.3.3.2.2)
3. apply setup rules to identify the principal setup to be used; (refer to chapter 9.3.3.2.3)
4. select a fixturing strategy;(refer to chapter 9.3.3.2.4)
5. remove any duplicate features or setups.(refer to chapter 9.3.3.2.5)

### 6.2.12 Find Poss.Tools - Option 8

This option represents the *Evaluate Features* activity explained in chapter 9.3.3. It takes as input the setup sequence which has been generated by the last option and performs the following functions:



1. Extract the Operation data associated with each feature in each setup. (refer to chapter 9.3.3).
2. Call tool selection procedure for each Operation. This procedure represents the *Identify Cutting Tools* and the *Calculate Processing Data* activities described in chapter 9.3.3. The relevant software listings for these are provided in appendix 5.5 and 5.6.

The output from option 8 is a list of possible tools, for each Operation, and the cutting parameters to be used with each. In cases where no tools are found by the software then the user is prompted, provided with data describing constraints on the Operation, and asked to select a tool from a listing provided. If there are no suitable tools then the user must abandon the planning session until a suitable tool can be entered into the tooling database.

Once the procedure has been completed successfully the setup plans are finalised, but the Operation plans are not. They still require to be sequenced and specific tools to be selected.

#### 6.2.13 Sequence Ops - Option 9

This option fulfils the *Sequence Operations* activity as described in chapter 9.4. Each setup is taken in turn, the Operations and tooling are assessed, using the sequence operations procedure listed in appendix 5.7, and an Operation plan is generated for each setup. At the end of the procedure the final setup and operation plans are output to the screen. A file is also generated using the name of the component in the model, and adding ".plan" to the name. This file contains the setup plans and Operation plans for each setup. The machining instruction reproduced, in appendix 7.5, illustrate the contents of this file for the Glacier workpiece.

#### 6.2.14 Find M/c Setups - Option 10

This option fulfils the first part of the *Extract Machine Control Code* activity as described in chapter 9.5. In option 10 the user is asked to specify the approximate position, of each fixture datum, relative to the machine table zero position. This data is then used, when generating the part programs, to control probing routines which locate the actual position of the fixture datum for each setup, and set fixture offset values in the machine controller accordingly.

#### 6.2.15 Extract NC Code - Option 11

This option fulfils the second part of the *Extract Machine Control Code* activity described in chapter 9.5. In option 11, the data produced at the end of options 9 and 10 are used to produce part programs for each setup, for the GE2000 controller on the Wadkin V4-6 machining centre. An extract of the ADA code used is documented in appendix 5. Files are automatically generate to store the part programs produced. File names use the component name, the setup number and ".prog" to distinguish the relevant part programs. For example, the file name for the part program for the first setup of a component named "test" would be "test\_setno1.prog". The part programs produced for the Glacier workpiece are documented in appendix 7.5

The part programs can then be downloaded to the GE2000 controller and used to machine the component. The part program uses the tool number in the tooling database as a "T code" in the part program. This means that either the tools must be loaded into the carousel to suit the database numbers, or the T codes in the part programs must be edited to suit the tool pockets being used in the carousel. In the experimental software there are less than twenty tools in the database, which means that the former option can be used as the Wad-

kin V4-6 being used in the experiments had a 20 pocket carousel. However, when a more realistic range of tools is covered there will be a need to relate the T codes to the appropriate pocket numbers in the carousel.

#### **6.2.16 Draw SDSM - Option 12**

This option calls a graphics utility to draw the SDSM model. On selecting this option the user is asked to specify the graphics device being used:

Which graphics display device type is to be used?

- 1) Autograph/Datatype X5A
- 2) Autograph/Datatype XK-1
- 3) Pericom M600
- 4) Non graphics terminal
- 5) Sun Workstation
- 6) RasterTech

Please select graphics display device type (1-6) :

Option 5 should be selected when working on a SUN workstation. A window will then be opened and a representation of the SDSM drawn.

#### **6.2.17 Remove drawing - Option 13**

This option is another graphics utility which removes the SDSM graphics window.

#### **6.2.18 Dims and Tols - Option 14**

This is the option which was being used to explore links to the Relationship Graph, as described in chapter 8. This option is inoperative.

#### **6.2.19 Put back data - Option 15**

This option takes the data which has been generated by the Machine Planner and populates the setup activity part of the PDM. Before this can be done it is necessary to move to the node. The command "uuuuuu" moves from the {E3 solid} node back up to the {definition} node. The commands "'setup activity" and "N" will move to the {setup activity} node. Option 15 can then be selected to put data back into the structure.



## APPENDIX 7

## MACHINING INSTRUCTIONS AND PART PROGRAMS GENERATED FOR TESTPIECES

### 7.1 INTRODUCTION

This appendix documents a selection of the machining instructions and part programs which were produced in the course of the research. Machining instructions which were generated for various instances of the individual features represented in the Product Model environment are provided. The machining instructions produced from the three variants of the bolster plate, used in the exploration of queries on Fixturing Strategies described in chapter 9, are provided. This illustrates the resulting setup and operation plans produced for each Fixturing Strategy type. Finally the machining instructions and part programs produced for the Glacier testpiece, are documented.

### 7.2 INDIVIDUAL FEATURES

#### 7.2.1 A Closed Pocket

The following machining instructions were generated for a Closed Pocket with Parameters: length 25 mm, width 20 mm, depth 5 mm, corner radius 7 mm, fillet radius 1 mm, and Ra 2.5 microns.

#### Possible Tools:

For ROUGHING, clear\_pocket TOP POCKET are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
10.	12.00 mm	SLOT_DRILL	EPEA161204FR	1568 88 (down_slotting) 1568 175 (across_slotting)
11.	8.00 mm	SLOT_DRILL	EPEA110804FR	2351 122 (down_slotting) 2351 244 (across_slotting)
12.	4.00 mm	SLOT_DRILL	EPEA070404FR	6016 240 (down_slotting) 6016 481 (across_slotting)

For ROUGHING, profile\_pocket TOP POCKET are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
10.	12.00 mm	SLOT_DRILL	EPEA161204FR	1756 197
11.	8.00 mm	SLOT_DRILL	EPEA110804FR	2432 253
12.	4.00 mm	SLOT_DRILL	EPEA070404FR	6216 497

For FINISHING, profile\_pocket TOP POCKET are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
7.	12.00 mm	END_MILL	MPFA0803PPFR	3017 103

#### Actual Tools:

11	8.00 mm	SLOT_DRILL	(rough clear and profile)
7	12.00 mm	END_MILL	(finish profile)

### 7.2.2 A Through Pocket

The following machining instructions were generated for a Through Pocket with Parameters: length 50 mm, width 40 mm, depth 20 mm, corner radius 15 mm, Ra 2.5 microns.

**Possible Tools:**

For ROUGHING, clear\_pocket LEFT THRU POCKET are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
8.	20.00 mm	SLOT_DRILL	EPEA222004FR	699 45 (down_slotting)
				699 89 (across_slotting)
9.	16.00 mm	SLOT_DRILL	EPEA191604FR	874 52 (down_slotting)
				874 105 (across_slotting)
10.	12.00 mm	SLOT_DRILL	EPEA161204FR	1346 75 (down_slotting)
				1346 151 (across_slotting)
11.	8.00 mm	SLOT_DRILL	EPEA110804FR	2236 116 (down_slotting)
				2236 232 (across_slotting)

For ROUGHING, profile\_pocket LEFT THRU POCKET are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
8.	20.00 mm	SLOT_DRILL	EPEA222004FR	740 95
9.	16.00 mm	SLOT_DRILL	EPEA191604FR	887 106
10.	12.00 mm	SLOT_DRILL	EPEA161204FR	1475 165
11.	8.00 mm	SLOT_DRILL	EPEA110804FR	2269 236

For FINISHING, profile\_pocket LEFT THRU POCKET are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
6.	24.90 mm	END_MILL	MPFA0803PPFR	2271 199

**Actual Tools:**

- 9 16.00 mm SLOT\_DRILL
- 6 24.90 mm END\_MILL

### 7.2.3 A Step

The following machining instructions were generated for a Step with parameters: length 300 mm, width 10 mm, depth 15 mm, Ra 2.5 microns

**Possible Tools:**

For ROUGHING, clear step FRONT STEP are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
4.	40.00 mm	END_MILL	TPKN1603PPFR	934 104
8.	20.00 mm	SLOT_DRILL	EPEA222004FR	974 125
9.	16.00 mm	SLOT_DRILL	EPEA191604FR	1162 139
10.	12.00 mm	SLOT_DRILL	EPEA161204FR	1465 164
11.	8.00 mm	SLOT_DRILL	EPEA110804FR	2320 241
12.	4.00 mm	SLOT_DRILL	EPEA070404FR	5611 449

For FINISHING\_FACENSIDE, finish step FRONT STEP are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
3.	50.00 mm	END_MILL	APFA1604PDFR	5346 831
5.	40.00 mm	END_MILL	TPAN1603PPFR	10000 177

- 6. 24.90 mm END\_MILL MPFA0803PPFR 6426 562
- 7. 12.00 mm END\_MILL MPFA0803PPFR 8795 299

**Actual Tools:**

- 8 20.00 mm SLOT\_DRILL
- 3 50.00 mm END\_MILL

**7.2.4 A Channel**

The following machining instructions were generated for a Channel with Parameters: length 150 mm, width 13 mm, depth 6 mm, ra 2.5 microns.

**Possible Tools:**

For ROUGHING, clear\_channel CHANNEL are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
10.	12.00 mm	SLOT_DRILL	EPEA161204FR	1491 167
11.	8.00 mm	SLOT_DRILL	EPEA110804FR	2236 232
12.	4.00 mm	SLOT_DRILL	EPEA070404FR	5723 458

For FINISHING\_FACENSIDE, face\_channel CHANNEL are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
7.	12.00 mm	END_MILL	MPFA0803PPFR	8795 299

**Actual Tools:**

- 10 12.00 mm SLOT\_DRILL
- 7 12.00 mm END\_MILL

**7.3 A Hole**

The following machining instructions were generated for a Hole with parameters: 12mm dia, 15mm depth

**Possible Tools:**

For ROUGHING, drill\_hole THRU HOLES\_12MM are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
14.	12.00 mm	DRILL	770	196

**Actual Tools:**

- 14 12.00 mm DRILL

**7.3.1 A Face**

The following machining instructions were generated for a Face with parameters: length 150 mm, width 110 mm, depth 3.25 mm Ra 2.5 microns

**Possible Tools:**

For ROUGHING, clear\_face BOTTOM FACE are :

NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
2.	63.00 mm	FACE_MILL	SCMM090408TR	550 330
4.	40.00 mm	END_MILL	TPKN1603PPFR	929 103

For FINISHING, finish\_face BOTTOM FACE are :



NO.	DIA.	DESCRIPTION.	TIP	SPEED/FEED (rpm/mm per min)
3.	50.00 mm	END_MILL APFA1604PDFR	1898	295
5.	40.00 mm	END_MILL TPAN1603PPFR	5032	89
6.	24.90 mm	END_MILL MPFA0803PPFR	2618	229

**Actual Tools:**

- 2 63.00 mm FACE\_MILL
- 3 50.00 mm END\_MILL

**7.4 THE BOLSTER PLATE**

**7.4.1 Machining Instructions - Through Clamping Strategy**

**Setup Information for setup number:1**

Number of tools required is 3  
 The cutting time (mins) is 6.60  
 The total tool path time (mins) is 7.46

The Tools Required Are :

TOOL NO. DESCRIPTION

- 2 63.00 mm FACE\_MILL
- 14 12.00 mm DRILL
- 3 50.00 mm END\_MILL

regions for primary location are:  
 FRONT FACE

regions for secondary location are:  
 TOP FACE

BACK FACE

regions for clamping are:  
 BACK FACE

datum for setup is FRONT\_LEFT corner,  
 and top surface of workpiece.

**Operation Sequence:**

clear\_face BOTTOM FACE, ROUGHING  
 using tool no. 2-- 63.00 mm FACE\_MILL 15 Tip SCMM090408TR  
 drill\_hole THRU HOLES\_12MM, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 drill\_hole THRU HOLES\_12M 2, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 drill\_hole THRU HOLES\_12M 3, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 drill\_hole THRU HOLES\_12M 4, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 finish\_face BOTTOM FACE, FINISHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR

**Setup Information for setup number 2**

Number of tools required is 2  
 The cutting time (mins) is 22.20  
 The total tool path time (mins) is 22.77

The Tools Required Are :

TOOL NO. DESCRIPTION

- 2 63.00 mm FACE\_MILL
- 3 50.00 mm END\_MILL

regions for primary location are:

FRONT FACE

regions for secondary location are:

BOTTOM FACE

BACK FACE

regions for clamping are:

BACK FACE

datum for setup is FRONT\_LEFT corner,  
and bottom surface of workpiece.

**Operation Sequence:**

clear\_face TOP FACE, ROUGHING

using tool no. 2-- 63.00 mm FACE\_MILL 15 Tip SCMM090408TR

finish\_face TOP FACE, FINISHING

using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR

**Setup Information for setup number 3**

Number of tools required is 6

The cutting time (mins) is 68.85

The total tool path time (mins) is 71.93

The Tools Required Are :

TOOL NO. DESCRIPTION

- 10 12.00 mm SLOT\_DRILL
- 4 40.00 mm END\_MILL
- 8 20.00 mm SLOT\_DRILL
- 11 8.00 mm SLOT\_DRILL
- 3 50.00 mm END\_MILL
- 7 12.00 mm END\_MILL

regions for primary location are:

BOTTOM FACE

regions for secondary location are:

THRU HOLES\_12MM

THRU HOLES\_12M 3

regions for clamping are:

THRU HOLES\_12MM

THRU HOLES\_12M 3

datum for setup is THRU HOLES\_12M 2

and top surface of workpiece.

**Operation Sequence:**

clear step FRONT STEP, ROUGHING

using tool no. 10-- 12.00 mm SLOT\_DRILL 0 Tip EPEA161204FR

profile BACK FACE, ROUGHING

using tool no. 4-- 40.00 mm END\_MILL 0 Tip TPKN1603PPFR

profile RIGHT FACE, ROUGHING

using tool no. 8-- 20.00 mm SLOT\_DRILL 0 Tip EPEA222004FR

profile FRONT FACE, ROUGHING

using tool no. 8-- 20.00 mm SLOT\_DRILL 0 Tip EPEA222004FR

profile LEFT FACE, ROUGHING

using tool no. 8-- 20.00 mm SLOT\_DRILL 0 Tip EPEA222004FR

clear\_pocket LEFT TOP POCKET, ROUGHING

using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR

clear\_pocket RIGHT TOP POCKET, ROUGHING

using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
clear\_pocket LEFT THRU POCKET, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
clear\_pocket RIGHT THRU POCKE, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
profile\_pocket LEFT TOP POCKET ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
profile\_pocket RIGHT TOP POCKET, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
profile\_pocket LEFT THRU POCKET, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
profile\_pocket RIGHT THRU POCKE, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
profile BACK FACE, FINISHING  
using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
profile RIGHT FACE, FINISHING  
using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
profile FRONT FACE, FINISHING  
using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
profile LEFT FACE, FINISHING  
using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
finish step FRONT STEP, FINISHING\_FACENSIDE  
using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
profile\_pocket LEFT TOP POCKET, FINISHING  
using tool no. 7-- 12.00 mm END\_MILL 0 Tip MPFA0803PPFR  
profile\_pocket RIGHT TOP POCKET, FINISHING  
using tool no. 7-- 12.00 mm END\_MILL 0 Tip MPFA0803PPFR  
profile\_pocket LEFT THRU POCKET, FINISHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
profile\_pocket RIGHT THRU POCKE, FINISHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR

**7.4.2 Machining Instructions - Down Clamping Strategy**

**Setup Information for setup number 1**

Number of tools required is 2  
The cutting time (mins) is 12.53  
The total tool path time (mins) is 13.02  
The Tools Required Are :

TOOL NO.	DESCRIPTION
2	63.00 mm FACE_MILL
3	50.00 mm END_MILL

regions for primary location are:

TOP FACE

regions for secondary location are:

BOTTOM FACE

FRONT FACE

regions for clamping are:

BOTTOM FACE

datum for setup is FRONT\_LEFT corner,

and top surface of workpiece.

**Operation Sequence:**



clear\_face BACK FACE, ROUGHING  
 using tool no. 2-- 63.00 mm FACE\_MILL 15 Tip SCMM090408TR  
 finish\_face BACK FACE, FINISHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR

**Setup Information for setup number 2**

Number of tools required is 4  
 The cutting time (mins) is 15.38  
 The total tool path time (mins) is 16.66  
 The Tools Required Are :

TOOL NO. DESCRIPTION

- 2 63.00 mm FACE\_MILL
- 8 20.00 mm SLOT\_DRILL
- 14 12.00 mm DRILL
- 3 50.00 mm END\_MILL

regions for primary location are:

BACK FACE

regions for secondary location are:

FRONT FACE

FRONT STEP

regions for clamping are:

FRONT FACE

datum for setup is FRONT\_LEFT corner,  
 and top surface of workpiece.

Operation Sequence:

clear\_face BOTTOM FACE, ROUGHING  
 using tool no. 2-- 63.00 mm FACE\_MILL 15 Tip SCMM090408TR  
 profile LEFT FACE, ROUGHING  
 using tool no. 8-- 20.00 mm SLOT\_DRILL 0 Tip EPEA222004FR  
 drill\_hole THRU HOLES\_12MM, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 finish\_face BOTTOM FACE, FINISHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
 profile LEFT FACE, FINISHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR

**Setup Information for setup number 3**

Number of tools required is 2  
 The cutting time (mins) is 2.80  
 The total tool path time (mins) is 3.25  
 The Tools Required Are :

TOOL NO. DESCRIPTION

- 2 63.00 mm FACE\_MILL
- 3 50.00 mm END\_MILL

regions for primary location are:

BOTTOM FACE

regions for secondary location are:

TOP FACE

BACK FACE

regions for clamping are:

TOP FACE

datum for setup is FRONT\_LEFT corner,  
 and bottom surface of workpiece.

**Operation Sequence:**

clear\_face FRONT FACE, ROUGHING  
 using tool no. 2-- 63.00 mm FACE\_MILL 15 Tip SCMM090408TR  
 finish\_face FRONT FACE, FINISHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR

**Setup Information for setup number 4**

Number of tools required is 3  
 The cutting time (mins) is 30.88  
 The total tool path time (mins) is 32.04

The Tools Required Are :

TOOL NO. DESCRIPTION

2 63.00 mm FACE\_MILL  
 8 20.00 mm SLOT\_DRILL  
 3 50.00 mm END\_MILL

regions for primary location are:

FRONT FACE

regions for secondary location are:

BACK FACE

BOTTOM FACE

regions for clamping are:

BACK FACE

datum for setup is FRONT\_LEFT corner,  
 and bottom surface of workpiece.

**Operation Sequence:**

clear\_face TOP FACE, ROUGHING  
 using tool no. 2-- 63.00 mm FACE\_MILL 15 Tip SCMM090408TR  
 profile RIGHT FACE, ROUGHING  
 using tool no. 8-- 20.00 mm SLOT\_DRILL 0 Tip EPEA222004FR  
 finish\_face TOP FACE, FINISHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
 profile RIGHT FACE, FINISHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR

**Setup Information for setup number 5**

Number of tools required is 5  
 The cutting time (mins) is 7.33  
 The total tool path time (mins) is 9.59

The Tools Required Are :

TOOL NO. DESCRIPTION

10 12.00 mm SLOT\_DRILL  
 11 8.00 mm SLOT\_DRILL  
 14 12.00 mm DRILL  
 3 50.00 mm END\_MILL  
 7 12.00 mm END\_MILL

regions for primary location are:

BOTTOM FACE

regions for secondary location are:

BACK FACE

BOTTOM FACE

regions for clamping are:

TOP FACE

datum for setup is THRU HOLES\_12MM

and top surface of workpiece.

**Operation Sequence:**

clear step FRONT STEP, ROUGHING  
 using tool no. 10-- 12.00 mm SLOT\_DRILL 0 Tip EPEA161204FR  
 clear\_pocket LEFT TOP POCKET, ROUGHING  
 using tool no. 10-- 12.00 mm SLOT\_DRILL 0 Tip EPEA161204FR  
 clear\_pocket RIGHT TOP POCKET, ROUGHING  
 using tool no. 10-- 12.00 mm SLOT\_DRILL 0 Tip EPEA161204FR  
 clear\_pocket LEFT THRU POCKET, ROUGHING  
 using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
 clear\_pocket RIGHT THRU POCCKE, ROUGHING  
 using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
 profile\_pocket LEFT TOP POCKET, ROUGHING  
 using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
 profile\_pocket RIGHT TOP POCKET, ROUGHING  
 using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
 profile\_pocket LEFT THRU POCKET, ROUGHING  
 using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
 profile\_pocket RIGHT THRU POCCKE, ROUGHING  
 using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
 drill\_hole THRU HOLES\_12M 2, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 drill\_hole THRU HOLES\_12M 3, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 drill\_hole THRU HOLES\_12M 4, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 finish step FRONT STEP, FINISHING\_FACENSIDE  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
 profile\_pocket LEFT TOP POCKET, FINISHING  
 using tool no. 7-- 12.00 mm END\_MILL 0 Tip MPFA0803PPFR  
 profile\_pocket RIGHT TOP POCKET, FINISHING  
 using tool no. 7-- 12.00 mm END\_MILL 0 Tip MPFA0803PPFR  
 profile\_pocket LEFT THRU POCKET, FINISHING  
 using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
 profile\_pocket RIGHT THRU POCCKE, FINISHING  
 using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR

**7.4.3 Machining Instructions - Side Clamping Strategy**

**Setup Information for setup number 1**

Number of tools required is 2  
 The cutting time (mins) is 12.53  
 The total tool path time (mins) is 13.02  
 The Tools Required Are :

TOOL NO.	DESCRIPTION
2	63.00 mm FACE_MILL
3	50.00 mm END_MILL

regions for primary location are:

TOP FACE

regions for secondary location are:

BOTTOM FACE

FRONT FACE



regions for clamping are:

**BOTTOM FACE**

datum for setup is **FRONT\_LEFT** corner,  
and top surface of workpiece.

**Operation Sequence:**

clear\_face **BACK FACE, ROUGHING**

using tool no. 2-- 63.00 mm **FACE\_MILL** 15 Tip **SCMM090408TR**

finish\_face **BACK FACE, FINISHING**

using tool no. 3-- 50.00 mm **END\_MILL** 0 Tip **APFA1604PDFR**

**Setup Information for setup number 2**

Number of tools required is 3

The cutting time (mins) is 15.12

The total tool path time (mins) is 16.17

The Tools Required Are :

**TOOL NO. DESCRIPTION**

2 63.00 mm **FACE\_MILL**

8 20.00 mm **SLOT\_DRILL**

3 50.00 mm **END\_MILL**

regions for primary location are:

**BACK FACE**

regions for secondary location are:

**FRONT FACE**

**FRONT STEP**

regions for clamping are:

**FRONT FACE**

datum for setup is **FRONT\_LEFT** corner,  
and top surface of workpiece.

**Operation Sequence:**

clear\_face **BOTTOM FACE, ROUGHING**

using tool no. 2-- 63.00 mm **FACE\_MILL** 15 Tip **SCMM090408TR**

profile **LEFT FACE, ROUGHING**

using tool no. 8-- 20.00 mm **SLOT\_DRILL** 0 Tip **EPEA222004FR**

finish\_face **BOTTOM FACE, FINISHING**

using tool no. 3-- 50.00 mm **END\_MILL** 0 Tip **APFA1604PDFR**

profile **LEFT FACE, FINISHING**

using tool no. 3-- 50.00 mm **END\_MILL** 0 Tip **APFA1604PDFR**

**Setup Information for setup number 3**

Number of tools required is 2

The cutting time (mins) is 2.80

The total tool path time (mins) is 3.25

The Tools Required Are :

**TOOL NO. DESCRIPTION**

2 63.00 mm **FACE\_MILL**

3 50.00 mm **END\_MILL**

regions for primary location are:

**BOTTOM FACE**

regions for secondary location are:

**TOP FACE**

**BACK FACE**

regions for clamping are:

**TOP FACE**

datum for setup is FRONT\_LEFT corner,  
and bottom surface of workpiece.

**Operation Sequence:**

clear\_face FRONT FACE, ROUGHING  
using tool no. 2-- 63.00 mm FACE\_MILL 15 Tip SCMM090408TR  
finish\_face FRONT FACE, FINISHING  
using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR

**Setup Information for setup number 4**

Number of tools required is 7  
The cutting time (mins) is 38.65  
The total tool path time (mins) is 41.61

The Tools Required Are :

TOOL NO. DESCRIPTION

2	63.00 mm FACE_MILL
10	12.00 mm SLOT_DRILL
8	20.00 mm SLOT_DRILL
11	8.00 mm SLOT_DRILL
14	12.00 mm DRILL
3	50.00 mm END_MILL
7	12.00 mm END_MILL

regions for primary location are:

FRONT FACE

regions for secondary location are:

BACK FACE

BOTTOM FACE

regions for clamping are:

BACK FACE

datum for setup is FRONT\_LEFT corner,  
and bottom surface of workpiece.

**Operation Sequence:**

clear\_face TOP FACE, ROUGHING  
using tool no. 2-- 63.00 mm FACE\_MILL 15 Tip SCMM090408TR  
clear\_step FRONT STEP, ROUGHING  
using tool no. 10-- 12.00 mm SLOT\_DRILL 0 Tip EPEA161204FR  
profile RIGHT FACE, ROUGHING  
using tool no. 8-- 20.00 mm SLOT\_DRILL 0 Tip EPEA222004FR  
clear\_pocket LEFT TOP POCKET, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
clear\_pocket RIGHT TOP POCKET, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
clear\_pocket LEFT THRU POCKET, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
clear\_pocket RIGHT THRU POCKE, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
profile\_pocket LEFT TOP POCKET, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
profile\_pocket RIGHT TOP POCKET, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
profile\_pocket LEFT THRU POCKET, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
profile\_pocket RIGHT THRU POCKE, ROUGHING  
using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR

drill\_hole THRU HOLES\_12MM, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 drill\_hole THRU HOLES\_12M 2, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 drill\_hole THRU HOLES\_12M 3, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 drill\_hole THRU HOLES\_12M 4, ROUGHING  
 using tool no. 14-- 12.00 mm DRILL  
 finish\_face TOP FACE, FINISHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
 profile RIGHT FACE, FINISHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
 finish\_step FRONT STEP, FINISHING\_FACENSIDE  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APFA1604PDFR  
 profile\_pocket LEFT TOP POCKET, FINISHING  
 using tool no. 7-- 12.00 mm END\_MILL 0 Tip MPFA0803PPFR  
 profile\_pocket RIGHT TOP POCKET, FINISHING  
 using tool no. 7-- 12.00 mm END\_MILL 0 Tip MPFA0803PPFR  
 profile\_pocket LEFT THRU POCKET, FINISHING  
 using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR  
 profile\_pocket RIGHT THRU POCCKE, FINISHING  
 using tool no. 11-- 8.00 mm SLOT\_DRILL 0 Tip EPEA110804FR

## 7.5 THE GLACIER TESTPIECE

### 7.5.1 Machining Instructions

#### Setup Information for setup number:1

Number of tools required is 3  
 The cutting time (mins) is 8.34  
 The total tool path time (mins) is 9.21

The Tools Required Are :

TOOL NO. DESCRIPTION

3	50.00 mm END_MILL
15	10.00 mm DRILL
2	63.00 mm FACE_MILL

regions for primary location are:

FRONT FACE

regions for secondary location are:

TOP FACE

BACK FACE

regions for clamping are:

BACK FACE

datum for setup is FRONT\_LEFT corner,

and top surface of workpiece.

Operation Sequence:

clear\_face BOTTOM FACE, ROUGHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APMA1604PDFR  
 drill\_hole THRU\_HOLES1, ROUGHING  
 using tool no. 15-- 10.00 mm DRILL  
 drill\_hole THRU\_HOLES2, ROUGHING



using tool no. 15-- 10.00 mm DRILL  
 drill\_hole THRU\_HOLES3, ROUGHING  
 using tool no. 15-- 10.00 mm DRILL  
 drill\_hole THRU\_HOLES4, ROUGHING  
 using tool no. 15-- 10.00 mm DRILL  
 finish\_face BOTTOM FACE, FINISHING  
 using tool no. 2-- 63.00 mm FACE\_MILL 15 Tip SCFM09048TR

**Setup Information for setup number:2**

Number of tools required is 2  
 The cutting time (mins) is 11.95  
 The total tool path time (mins) is 12.47

The Tools Required Are :

TOOL NO. DESCRIPTION

- 3 50.00 mm END\_MILL
- 2 63.00 mm FACE\_MILL

regions for primary location are:

FRONT FACE

regions for secondary location are:

BOTTOM FACE

BACK FACE

regions for clamping are:

BACK FACE

datum for setup is FRONT\_LEFT corner,

and bottom surface of workpiece.

Operation Sequence:

clear\_face TOP FACE, ROUGHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APMA1604PDFR  
 finish\_face TOP FACE, FINISHING  
 using tool no. 2-- 63.00 mm FACE\_MILL 15 Tip SCFM09048TR

**Setup Information for setup number:3**

Number of tools required is 5  
 The cutting time (mins) is 31.86  
 The total tool path time (mins) is 33.77

The Tools Required Are :

TOOL NO. DESCRIPTION

- 3 50.00 mm END\_MILL
- 10 12.00 mm SLOT\_DRILL
- 18 3.00 mm DRILL
- 8 20.00 mm SLOT\_DRILL
- 7 12.00 mm END\_MILL

regions for primary location are:

BOTTOM FACE

regions for secondary location are:

THRU\_HOLES1

THRU\_HOLES3

regions for clamping are:

THRU\_HOLES1

THRU\_HOLES3

datum for setup is THRU\_HOLES2

and top surface of workpiece.

Operation Sequence:

profile BACK FACE, ROUGHING

using tool no. 3-- 50.00 mm END\_MILL 0 Tip APMA1604PDFR  
 profile RIGHT FACE, ROUGHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APMA1604PDFR  
 profile FRONT FACE, ROUGHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APMA1604PDFR  
 profile LEFT FACE, ROUGHING  
 using tool no. 3-- 50.00 mm END\_MILL 0 Tip APMA1604PDFR  
 clear\_channel CHANNEL, ROUGHING  
 using tool no. 10-- 12.00 mm SLOT\_DRILL 0 Tip EPEA161204FR  
 drill\_hole BLIND\_HOLES\_3MM, ROUGHING  
 using tool no. 18-- 3.00 mm DRILL  
 drill\_hole BLIND\_HOLES\_3M 2, ROUGHING  
 using tool no. 18-- 3.00 mm DRILL  
 drill\_hole BLIND\_HOLES\_3M 3, ROUGHING  
 using tool no. 18-- 3.00 mm DRILL  
 drill\_hole BLIND\_HOLES\_3M 4, ROUGHING  
 using tool no. 18-- 3.00 mm DRILL  
 profile BACK FACE, FINISHING  
 using tool no. 8-- 20.00 mm SLOT\_DRILL 0 Tip EPFA222004FR  
 profile RIGHT FACE, FINISHING  
 using tool no. 8-- 20.00 mm SLOT\_DRILL 0 Tip EPFA222004FR  
 profile FRONT FACE, FINISHING  
 using tool no. 8-- 20.00 mm SLOT\_DRILL 0 Tip EPFA222004FR  
 profile LEFT FACE, FINISHING  
 using tool no. 8-- 20.00 mm SLOT\_DRILL 0 Tip EPFA222004FR  
 face\_channel CHANNEL, FINISHING\_FACENSIDE  
 using tool no. 7-- 12.00 mm END\_MILL 0 Tip MPFA0803PPFR

### 7.5.2 Part Program for Setup 1.

```

n0010 (id,prog,glaci1,15/MAR/1990 at 11:49)
n0020 !
n0030 ! set_datum
n0040 !
n0050 m06 t1
n0060 d1
n0070 p93 =20.000 ! safe rapid height
n0080 p87 =0.000 ! z coordinate of corner
n0090 p86 =160.000 ! y coordinate of corner
n0100 p85 =395.000 ! x coordinate of corner
n0110 p84 =1 ! corner type
n0120 p83 =1 ! fixture offset number
n0130 (gsub,extcm)
n0140 !
n0150 ! clear_face
n0160 !
n0170 m06 l3
n0180 d3
n0190 e1
n0200 s1000 m03
n0210 g53
n0220 g56 x80.000 y62.500 z-3.000
n0230 g00 x0.000 y0.000 z23.000
n0240 g00 x-110.000 y64.167 z23.000
n0250 g00 x-110.000 y64.167 z0.000
n0260 g01 x110.000 y64.167 z0.000 f310.946 s378.339
n0270 g00 x110.000 y-64.167 z0.000
  
```

n0280 g01 x-110.000 y-64.167 z0.000  
n0290 g00 x-110.000 y40.833 z0.000  
n0300 g01 x110.000 y40.833 z0.000  
n0310 g00 x110.000 y-40.833 z0.000  
n0320 g01 x-110.000 y-40.833 z0.000  
n0330 g00 x-110.000 y12.500 z0.000  
n0340 g01 x110.000 y12.500 z0.000  
n0350 g00 x110.000 y-12.500 z0.000  
n0360 g01 x-110.000 y-12.500 z0.000  
n0370 g00 x-110.000 y-12.500 z23.000  
n0380 !  
n0390 ! drill\_hole  
n0400 !  
n0410 m06 t15  
n0420 d15  
n0430 e1  
n0440 s1000 m03  
n0450 g53  
n0460 g56 x20.750 y96.250 z-3.250  
n0470 g00 x-110.000 y-12.500 z23.250  
n0480 g00 x0.000 y0.000 z23.250  
n0490 g00 x0.000 y0.000 z3.000  
n0500 g01 z-23.000 f235.510 s923.566  
n0510 g01 z3.000  
n0520 g00 x0.000 y0.000 z23.250  
n0530 !  
n0540 ! drill\_hole  
n0550 !  
n0560 e1  
n0570 s1000 m03  
n0580 g53  
n0590 g56 x20.750 y32.250 z-3.250  
n0600 g00 x0.000 y0.000 z23.250  
n0610 g00 x0.000 y0.000 z23.250  
n0620 g00 x0.000 y0.000 z3.000  
n0630 g01 z-23.000 f235.510 s923.566  
n0640 g01 z3.000  
n0650 g00 x0.000 y0.000 z23.250  
n0660 !  
n0670 ! drill\_hole  
n0680 !  
n0690 e1  
n0700 s1000 m03  
n0710 g53  
n0720 g56 x142.750 y32.250 z-3.250  
n0730 g00 x0.000 y0.000 z23.250  
n0740 g00 x0.000 y0.000 z23.250  
n0750 g00 x0.000 y0.000 z3.000  
n0760 g01 z-23.000 f235.510 s923.566  
n0770 g01 z3.000  
n0780 g00 x0.000 y0.000 z23.250  
n0790 !  
n0800 ! drill\_hole  
n0810 !  
n0820 e1  
n0830 s1000 m03  
n0840 g53  
n0850 g56 x142.750 y96.250 z-3.250  
n0860 g00 x0.000 y0.000 z23.250  
n0870 g00 x0.000 y0.000 z23.250  
n0880 g00 x0.000 y0.000 z3.000  
n0890 g01 z-23.000 f235.510 s923.566  
n0900 g01 z3.000



```

n0910 g00 x0.000 y0.000 z23.250
n0920 !
n0930 ! finish_face
n0940 !
n0950 m06 t2
n0960 d2
n0970 e1
n0980 s1000 m03
n0990 g53
n1000 g56 x80.000 y62.500 z-3.250
n1010 g00 x0.000 y0.000 z23.250
n1020 g00 x-117.800 y58.025 z23.250
n1030 g00 x-117.800 y58.025 z0.000
n1040 g01 x117.800 y58.025 z0.000 f293.596 s3083.993
n1050 g00 x117.800 y-58.025 z0.000
n1060 g01 x-117.800 y-58.025 z0.000
n1070 g00 x-117.800 y15.750 z0.000
n1080 g01 x117.800 y15.750 z0.000
n1090 g00 x117.800 y-15.750 z0.000
n1100 g01 x-117.800 y-15.750 z0.000
n1110 g00 x-117.800 y-15.750 z23.250
n1120 g53
n1130 c0
n1140 d0
n1150 m06 t2
n1160 g00 x450 y450
n1170 m30
n9999 (end,prog)

```

### 7.5.3 Part Program for Setup 2.

```

n0010 (id,prog,glaci2,15/MAR/1990 at 11:51)
n0020 !
n0030 ! set_datum_x
n0040 !
n0050 m06 t1
n0060 d1
n0070 p93 =46.000 ! safe rapid height
n0080 p92 =10 ! probing length
n0090 p91 =-10.000 ! z coordinate
n0100 p90 =170.000 ! y coordinate
n0110 p89 =395.000 ! x coordinate
n0120 p88 = +1 ! probing direction
n0130 p83 =2 ! fixture offset number
n0140 (gsub,xdatum)
n0150 !
n0160 ! set_datum_y
n0170 !
n0180 p93 =46.000 ! safe rapid height
n0190 p92 =10 ! probing length
n0200 p91 =-10.000 ! z coordinate
n0210 p90 =160.000 ! y coordinate
n0220 p89 =405.000 ! x coordinate
n0230 p88 = +1 ! probing direction
n0240 p83 =2 ! fixture offset number
n0250 (gsub,ydatum)
n0260 !
n0270 ! set_datum_z
n0280 !
n0290 p93 =46.000 ! safe rapid height
n0300 p92 =10 ! probing length
n0310 p91 =-26.000 ! z coordinate

```

n0320 p90 =150.000 ! y coordinate  
n0330 p89 =435.000 ! x coordinate  
n0340 p83 =2 ! fixture offset number  
n0350 (gsub,zdatum)  
n0360 !  
n0370 ! clear\_face  
n0380 !  
n0390 m06 l3  
n0400 d3  
n0410 e2  
n0420 s1000 m03  
n0430 g53  
n0440 g56 x80.000 y62.500 z16.250  
n0450 g00 x0.000 y0.000 z29.750  
n0460 g00 x-110.000 y64.167 z29.750  
n0470 g00 x-110.000 y64.167 z3.250  
n0480 g01 x110.000 y64.167 z3.250 f302.183 s367.676  
n0490 g00 x110.000 y-64.167 z3.250  
n0500 g01 x-110.000 y-64.167 z3.250  
n0510 g00 x-110.000 y40.833 z3.250  
n0520 g01 x110.000 y40.833 z3.250  
n0530 g00 x110.000 y-40.833 z3.250  
n0540 g01 x-110.000 y-40.833 z3.250  
n0550 g00 x-110.000 y12.500 z3.250  
n0560 g01 x110.000 y12.500 z3.250  
n0570 g00 x110.000 y-12.500 z3.250  
n0580 g01 x-110.000 y-12.500 z3.250  
n0590 g00 x-110.000 y64.167 z3.250  
n0600 g00 x-110.000 y64.167 z0.000  
n0610 g01 x110.000 y64.167 z0.000  
n0620 g00 x110.000 y-64.167 z0.000  
n0630 g01 x-110.000 y-64.167 z0.000  
n0640 g00 x-110.000 y40.833 z0.000  
n0650 g01 x110.000 y40.833 z0.000  
n0660 g00 x110.000 y-40.833 z0.000  
n0670 g01 x-110.000 y-40.833 z0.000  
n0680 g00 x-110.000 y12.500 z0.000  
n0690 g01 x110.000 y12.500 z0.000  
n0700 g00 x110.000 y-12.500 z0.000  
n0710 g01 x-110.000 y-12.500 z0.000  
n0720 g00 x-110.000 y-12.500 z29.750  
n0730 !  
n0740 ! finish\_face  
n0750 !  
n0760 m06 l2  
n0770 d2  
n0780 e2  
n0790 s1000 m03  
n0800 g53  
n0810 g56 x80.000 y62.500 z16.000  
n0820 g00 x-110.000 y-12.500 z30.000  
n0830 g00 x-117.800 y58.025 z30.000  
n0840 g00 x-117.800 y58.025 z0.000  
n0850 g01 x117.800 y58.025 z0.000 f293.596 s3083.993  
n0860 g00 x117.800 y-58.025 z0.000  
n0870 g01 x-117.800 y-58.025 z0.000  
n0880 g00 x-117.800 y15.750 z0.000  
n0890 g01 x117.800 y15.750 z0.000  
n0900 g00 x117.800 y-15.750 z0.000  
n0910 g01 x-117.800 y-15.750 z0.000  
n0920 g00 x-117.800 y-15.750 z30.000  
n0930 g53  
n0940 c0

```

n0950 d0
n0960 m06 t2
n0970 g00 x450 y450
n0980 m30
n9999 (end,prog)

```

#### 7.5.4 Part Program for Setup 3.

```

n0010 (id,prog,glaci3,15/MAR/1990 at 11:55)
n0020 !
n0030 ! set_datum_bore
n0040 !
n0050 m06 t1
n0060 d1
n0070 p93 =20.000 ! safe rapid height
n0080 p87 =-5.000 ! z coordinate of corner
n0090 p86 =262.000 ! y coordinate of corner
n0100 p85 =78.500 ! x coordinate of corner
n0110 p84 =5.000 ! radius
n0120 p83 =3 ! fixture offset number
n0130 (gsub,bore)
n0140 !
n0150 ! profile
n0160 !
n0170 m06 t3
n0180 d3
n0190 e3
n0200 s1000 m03
n0210 g53
n0220 g56 x139.250 y32.250 z7.000
n0230 g00 x0.000 y0.000 z13.000
n0240 g00 x-190.000 y18.500 z13.000
n0250 g00 x-190.000 y18.500 z3.000
n0260 g00 x-190.000 y18.500 z-9.667
n0270 g01 x30.000 y18.500 z-9.667 f291.283 s297.433
n0280 g00 x30.000 y18.500 z3.000
n0290 g00 x-190.000 y18.500 z3.000
n0300 g00 x-190.000 y18.500 z3.000
n0310 g00 x-190.000 y18.500 z-19.333
n0320 g01 x30.000 y18.500 z-19.333
n0330 g00 x30.000 y18.500 z3.000
n0340 g00 x-190.000 y18.500 z3.000
n0350 g00 x-190.000 y18.500 z3.000
n0360 g00 x-190.000 y18.500 z-29.000
n0370 g01 x30.000 y18.500 z-29.000
n0380 g00 x30.000 y18.500 z13.000
n0390 !
n0400 ! profile
n0410 !
n0420 e3
n0430 s1000 m03
n0440 g53
n0450 g56 x139.250 y-92.750 z7.000
n0460 g00 y-30.000 x18.500 z13.000
n0470 g00 y155.000 x22.000 z13.000
n0480 g00 y155.000 x22.000 z3.000
n0490 g00 y155.000 x22.000 z-9.667
n0500 g01 y-30.000 x22.000 z-9.667 f345.844 s353.145
n0510 g00 y-30.000 x22.000 z3.000
n0520 g00 y155.000 x22.000 z3.000
n0530 g00 y155.000 x22.000 z3.000
n0540 g00 y155.000 x22.000 z-19.333

```



n0550 g01 y-30.000 x22.000 z-19.333  
n0560 g00 y-30.000 x22.000 z3.000  
n0570 g00 y155.000 x22.000 z3.000  
n0580 g00 y155.000 x22.000 z3.000  
n0590 g00 y155.000 x22.000 z-29.000  
n0600 g01 y-30.000 x22.000 z-29.000  
n0610 g00 y-30.000 x22.000 z13.000  
n0620 !  
n0630 ! profile  
n0640 !  
n0650 e3  
n0660 s1000 m03  
n0670 g53  
n0680 g56 x-20.750 y-92.750 z7.000  
n0690 g00 x-30.000 y-22.000 z13.000  
n0700 g00 x190.000 y-22.000 z13.000  
n0710 g00 x190.000 y-22.000 z3.000  
n0720 g00 x190.000 y-22.000 z-9.667  
n0730 g01 x-30.000 y-22.000 z-9.667 r345.844 s353.145  
n0740 g00 x-30.000 y-22.000 z3.000  
n0750 g00 x190.000 y-22.000 z3.000  
n0760 g00 x190.000 y-22.000 z3.000  
n0770 g00 x190.000 y-22.000 z-19.333  
n0780 g01 x-30.000 y-22.000 z-19.333  
n0790 g00 x-30.000 y-22.000 z3.000  
n0800 g00 x190.000 y-22.000 z3.000  
n0810 g00 x190.000 y-22.000 z3.000  
n0820 g00 x190.000 y-22.000 z-29.000  
n0830 g01 x-30.000 y-22.000 z-29.000  
n0840 g00 x-30.000 y-22.000 z13.000  
n0850 !  
n0860 ! profile  
n0870 !  
n0880 e3  
n0890 s1000 m03  
n0900 g53  
n0910 g56 x-20.750 y32.250 z7.000  
n0920 g00 y30.000 x-22.000 z13.000  
n0930 g00 y-155.000 x-18.500 z13.000  
n0940 g00 y-155.000 x-18.500 z3.000  
n0950 g00 y-155.000 x-18.500 z-9.667  
n0960 g01 y30.000 x-18.500 z-9.667 r291.283 s297.433  
n0970 g00 y30.000 x-18.500 z3.000  
n0980 g00 y-155.000 x-18.500 z3.000  
n0990 g00 y-155.000 x-18.500 z3.000  
n1000 g00 y-155.000 x-18.500 z-19.333  
n1010 g01 y30.000 x-18.500 z-19.333  
n1020 g00 y30.000 x-18.500 z3.000  
n1030 g00 y-155.000 x-18.500 z3.000  
n1040 g00 y-155.000 x-18.500 z3.000  
n1050 g00 y-155.000 x-18.500 z-29.000  
n1060 g01 y30.000 x-18.500 z-29.000  
n1070 g00 y30.000 x-18.500 z13.000  
n1080 !  
n1090 ! clear\_channel  
n1100 !  
n1110 m06 t10  
n1120 d10  
n1130 e3  
n1140 s1000 m03  
n1150 g53  
n1160 g56 x-14.000 y-38.250  
n1170 g00 x-30.000 y-18.500 z20.000

n1180 g00 x157.200 y6.500 z20.000  
n1190 g00 x157.200 y6.500 z3.000  
n1200 g00 x157.200 y6.500 z-5.750  
n1210 g01 x-7.200 y6.500 z-5.750 f166.917 s1491.438  
n1220 g00 x-7.200 y6.000 z-5.750  
n1230 g00 x-7.200 y6.000 z-5.750  
n1240 g01 x157.200 y6.000 z-5.750 f144.953 s1295.185  
n1250 g00 x157.200 y6.000 z3.000  
n1260 g00 x157.200 y6.000 z20.000  
n1270 !  
n1280 ! drill\_hole  
n1290 !  
n1300 m06 t18  
n1310 d18  
n1320 e3  
n1330 s1000 m03  
n1340 g53  
n1350 g56 x36.500 y-84.000  
n1360 g00 x157.200 y6.000 z20.000  
n1370 g00 x0.000 y0.000 z20.000  
n1380 g00 x0.000 y0.000 z3.000  
n1390 g01 z-11.000 f785.032 s3078.557  
n1400 g01 z3.000  
n1410 g00 x0.000 y0.000 z20.000  
n1420 !  
n1430 ! drill\_hole  
n1440 !  
n1450 e3  
n1460 s1000 m03  
n1470 g53  
n1480 g56 x36.500 y20.000  
n1490 g00 x0.000 y0.000 z20.000  
n1500 g00 x0.000 y0.000 z20.000  
n1510 g00 x0.000 y0.000 z3.000  
n1520 g01 z-11.000 f785.032 s3078.557  
n1530 g01 z3.000  
n1540 g00 x0.000 y0.000 z20.000  
n1550 !  
n1560 ! drill\_hole  
n1570 !  
n1580 e3  
n1590 s1000 m03  
n1600 g53  
n1610 g56 x85.500 y20.000  
n1620 g00 x0.000 y0.000 z20.000  
n1630 g00 x0.000 y0.000 z20.000  
n1640 g00 x0.000 y0.000 z3.000  
n1650 g01 z-11.000 f785.032 s3078.557  
n1660 g01 z3.000  
n1670 g00 x0.000 y0.000 z20.000  
n1680 !  
n1690 ! drill\_hole  
n1700 !  
n1710 e3  
n1720 s1000 m03  
n1730 g53  
n1740 g56 x85.500 y-84.000  
n1750 g00 x0.000 y0.000 z20.000  
n1760 g00 x0.000 y0.000 z20.000  
n1770 g00 x0.000 y0.000 z3.000  
n1780 g01 z-11.000 f785.032 s3078.557  
n1790 g01 z3.000  
n1800 g00 x0.000 y0.000 z20.000

n1810 !  
n1820 ! profile  
n1830 !  
n1840 m06 t8  
n1850 d8  
n1860 e3  
n1870 s1000 m03  
n1880 g53  
n1890 g56 x139.250 y25.750 z6.750  
n1900 g00 x0.000 y0.000 z13.250  
n1910 g00 x-172.000 y9.750 z13.250  
n1920 g00 x-172.000 y9.750 z3.000  
n1930 g00 x-172.000 y9.750 z-14.500  
n1940 g01 x12.000 y9.750 z-14.500 f67.568 s3328.109  
n1950 g00 x12.000 y9.750 z3.000  
n1960 g00 x-172.000 y9.750 z3.000  
n1970 g00 x-172.000 y9.750 z3.000  
n1980 g00 x-172.000 y9.750 z-29.000  
n1990 g01 x12.000 y9.750 z-29.000  
n2000 g00 x12.000 y9.750 z13.250  
n2010 !  
n2020 ! profile  
n2030 !  
n2040 e3  
n2050 s1000 m03  
n2060 g53  
n2070 g56 x136.250 y-92.750 z6.750  
n2080 g00 y-12.000 x9.750 z13.250  
n2090 g00 y137.000 x9.750 z13.250  
n2100 g00 y137.000 x9.750 z3.000  
n2110 g00 y137.000 x9.750 z-14.500  
n2120 g01 y-12.000 x9.750 z-14.500 f67.568 s3328.109  
n2130 g00 y-12.000 x9.750 z3.000  
n2140 g00 y137.000 x9.750 z3.000  
n2150 g00 y137.000 x9.750 z3.000  
n2160 g00 y137.000 x9.750 z-29.000  
n2170 g01 y-12.000 x9.750 z-29.000  
n2180 g00 y-12.000 x9.750 z13.250  
n2190 !  
n2200 ! profile  
n2210 !  
n2220 e3  
n2230 s1000 m03  
n2240 g53  
n2250 g56 x-20.750 y-89.750 z6.750  
n2260 g00 x-12.000 y-9.750 z13.250  
n2270 g00 x172.000 y-9.750 z13.250  
n2280 g00 x172.000 y-9.750 z3.000  
n2290 g00 x172.000 y-9.750 z-14.500  
n2300 g01 x-12.000 y-9.750 z-14.500 f67.568 s3328.109  
n2310 g00 x-12.000 y-9.750 z3.000  
n2320 g00 x172.000 y-9.750 z3.000  
n2330 g00 x172.000 y-9.750 z3.000  
n2340 g00 x172.000 y-9.750 z-29.000  
n2350 g01 x-12.000 y-9.750 z-29.000  
n2360 g00 x-12.000 y-9.750 z13.250  
n2370 !  
n2380 ! profile  
n2390 !  
n2400 e3  
n2410 s1000 m03  
n2420 g53  
n2430 g56 x-14.250 y32.250 z6.750

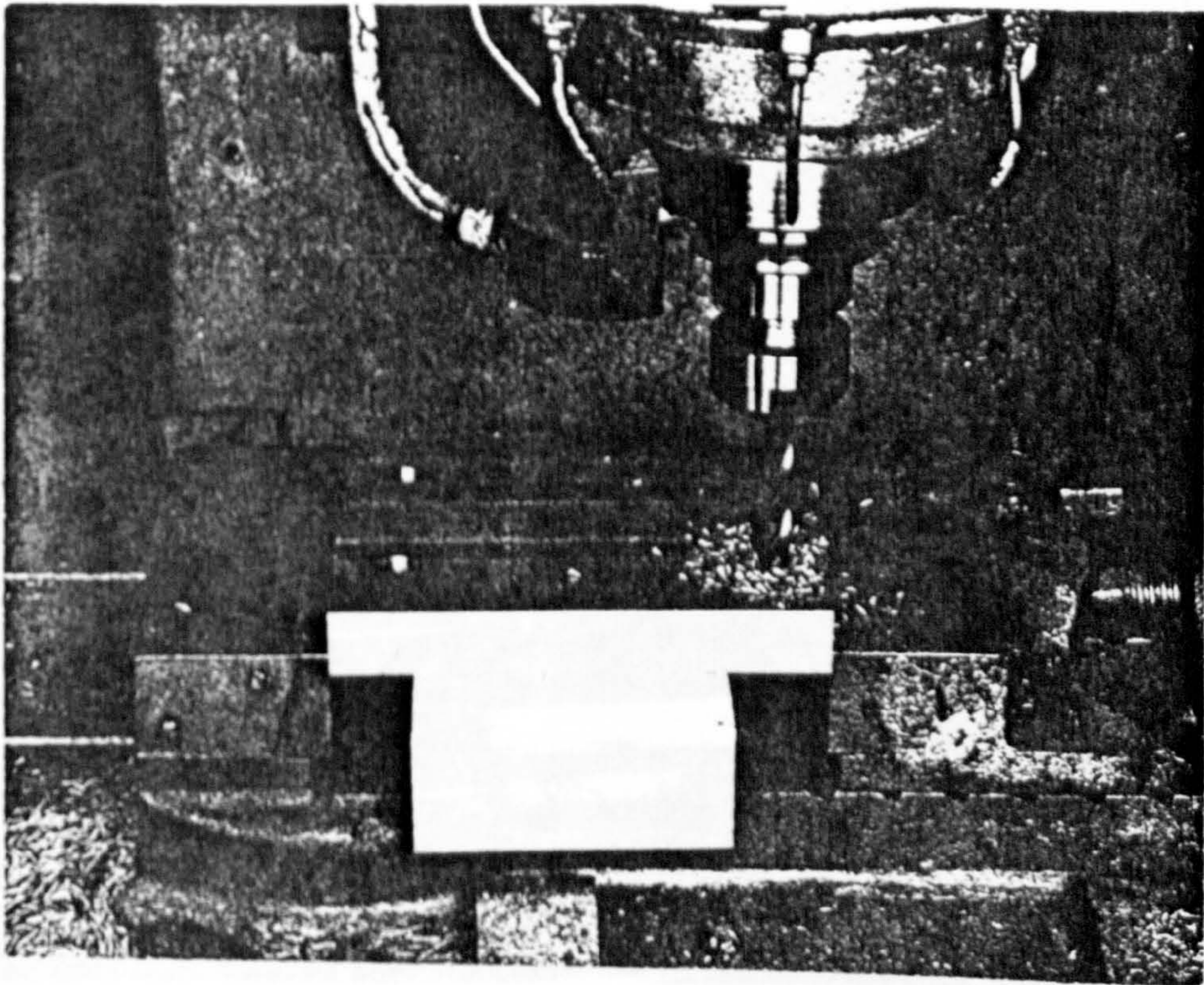


n2440 g00 y12.000 x-9.750 z13.250  
n2450 g00 y-137.000 x-9.750 z13.250  
n2460 g00 y-137.000 x-9.750 z3.000  
n2470 g00 y-137.000 x-9.750 z-14.500  
n2480 g01 y12.000 x-9.750 z-14.500 f67.568 s3328.109  
n2490 g00 y12.000 x-9.750 z3.000  
n2500 g00 y-137.000 x-9.750 z3.000  
n2510 g00 y-137.000 x-9.750 z3.000  
n2520 g00 y-137.000 x-9.750 z-29.000  
n2530 g01 y12.000 x-9.750 z-29.000  
n2540 g00 y12.000 x-9.750 z13.250  
n2550 !  
n2560 ! face\_channel  
n2570 !  
n2580 m06 i7  
n2590 d7  
n2600 e3  
n2610 s1000 m03  
n2620 g53  
n2630 g56 x-14.000 y-38.500  
n2640 g00 x-12.000 y-9.750 z20.000  
n2650 g00 x157.200 y7.000 z20.000  
n2660 g00 x157.200 y7.000 z-3.000  
n2670 g01 x-7.200 y7.000 z-3.000 f299.354 s4000  
n2680 g00 x-7.200 y7.000 z3.000  
n2690 g00 x157.200 y7.000 z3.000  
n2700 g00 x157.200 y7.000 z-6.000  
n2710 g01 x-7.200 y7.000 z-6.000  
n2720 g00 x-7.200 y6.000 z-6.000  
n2730 g00 x-7.200 y6.000 z-3.000  
n2740 g01 x157.200 y6.000 z-3.000  
n2750 g00 x157.200 y6.000 z3.000  
n2760 g00 x-7.200 y6.000 z3.000  
n2770 g00 x-7.200 y6.000 z-6.000  
n2780 g01 x157.200 y6.000 z-6.000  
n2790 g00 x157.200 y6.000 z20.000  
n2800 g53  
n2810 e0  
n2820 d0  
n2830 m06 i7  
n2840 g00 x450 y450  
n2850 m30  
n9999 (end.prog)

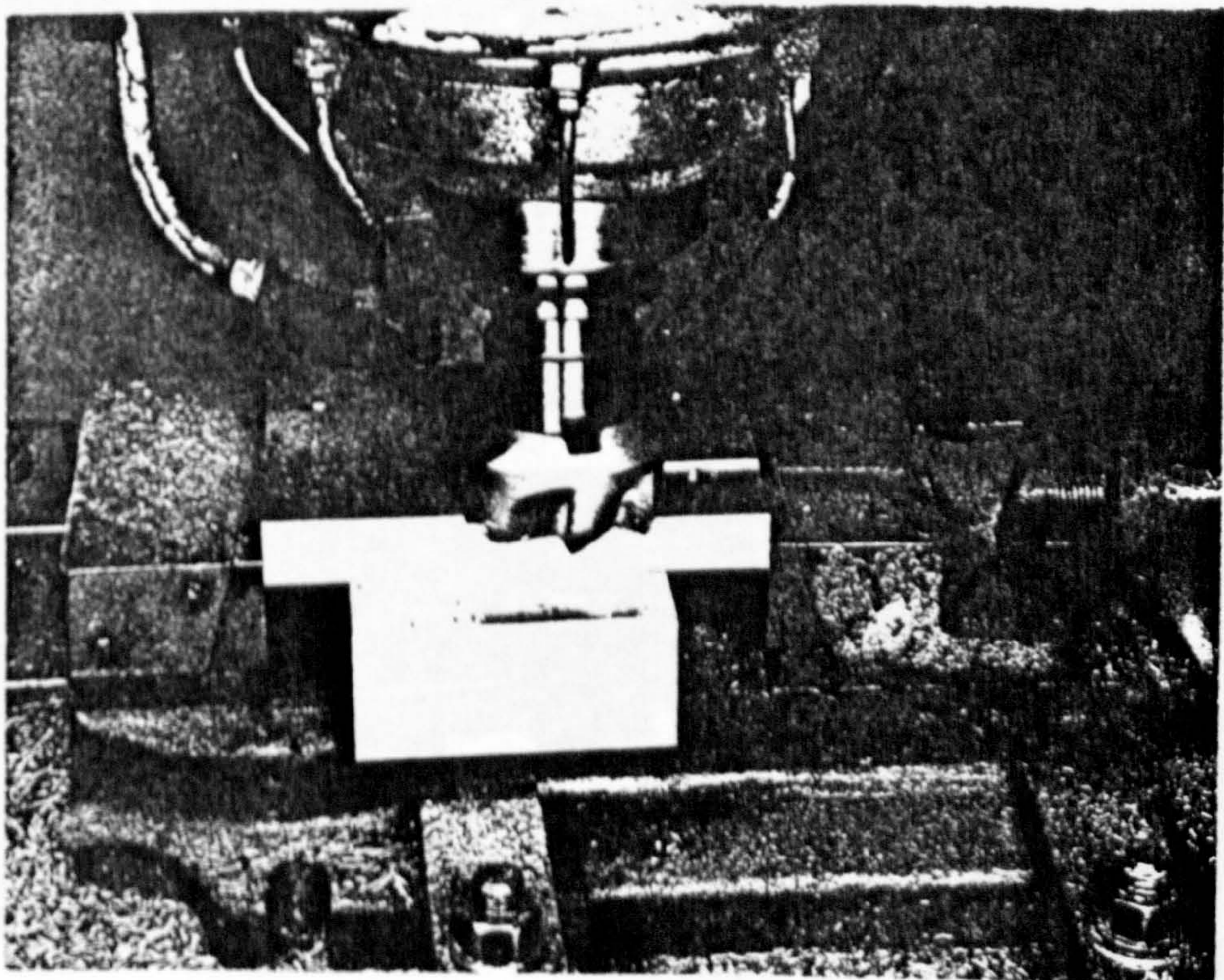


# MACHINING THE GLACIER WORKPIECE

(Example Operations)

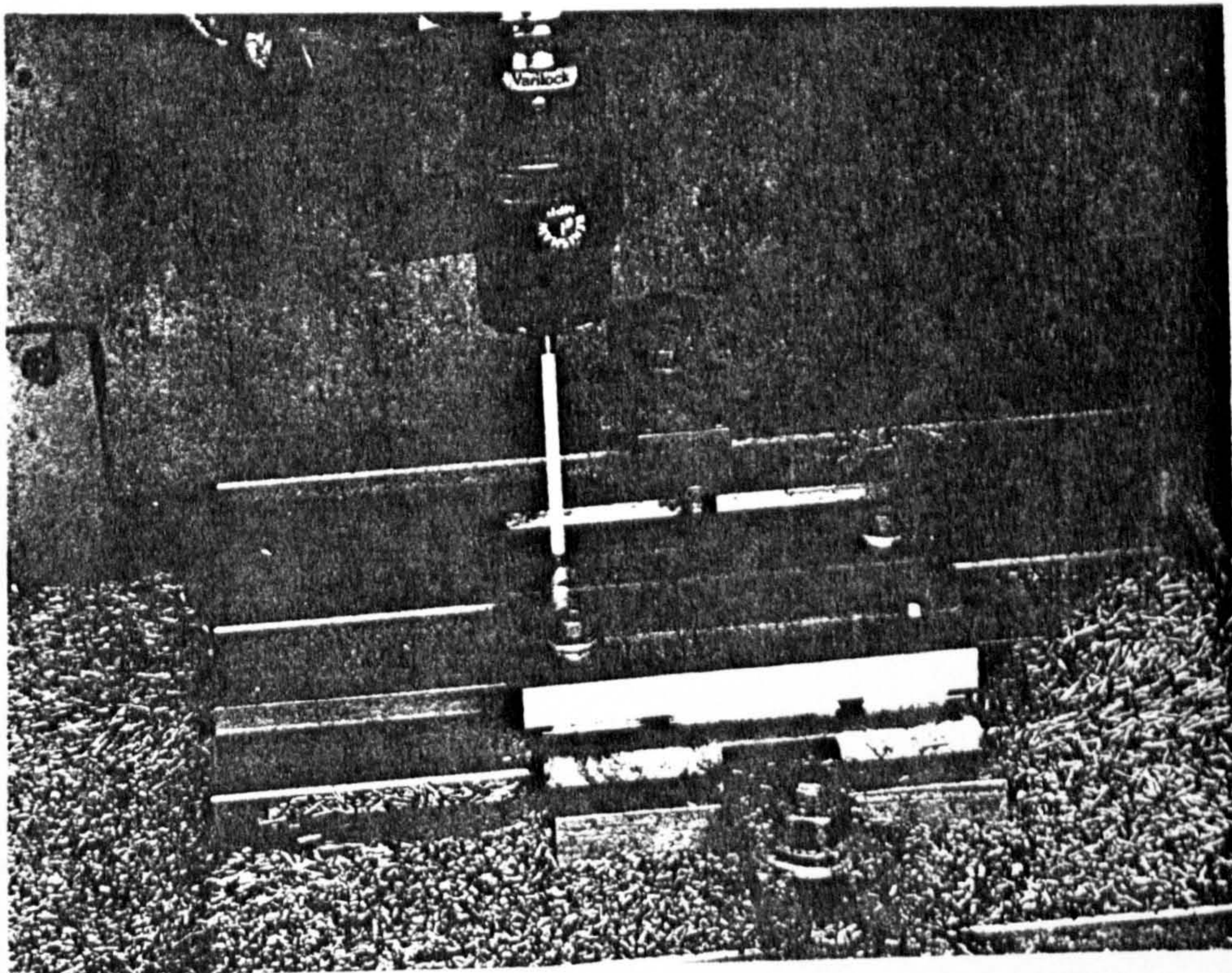


Setup 1 - Drilling Operation

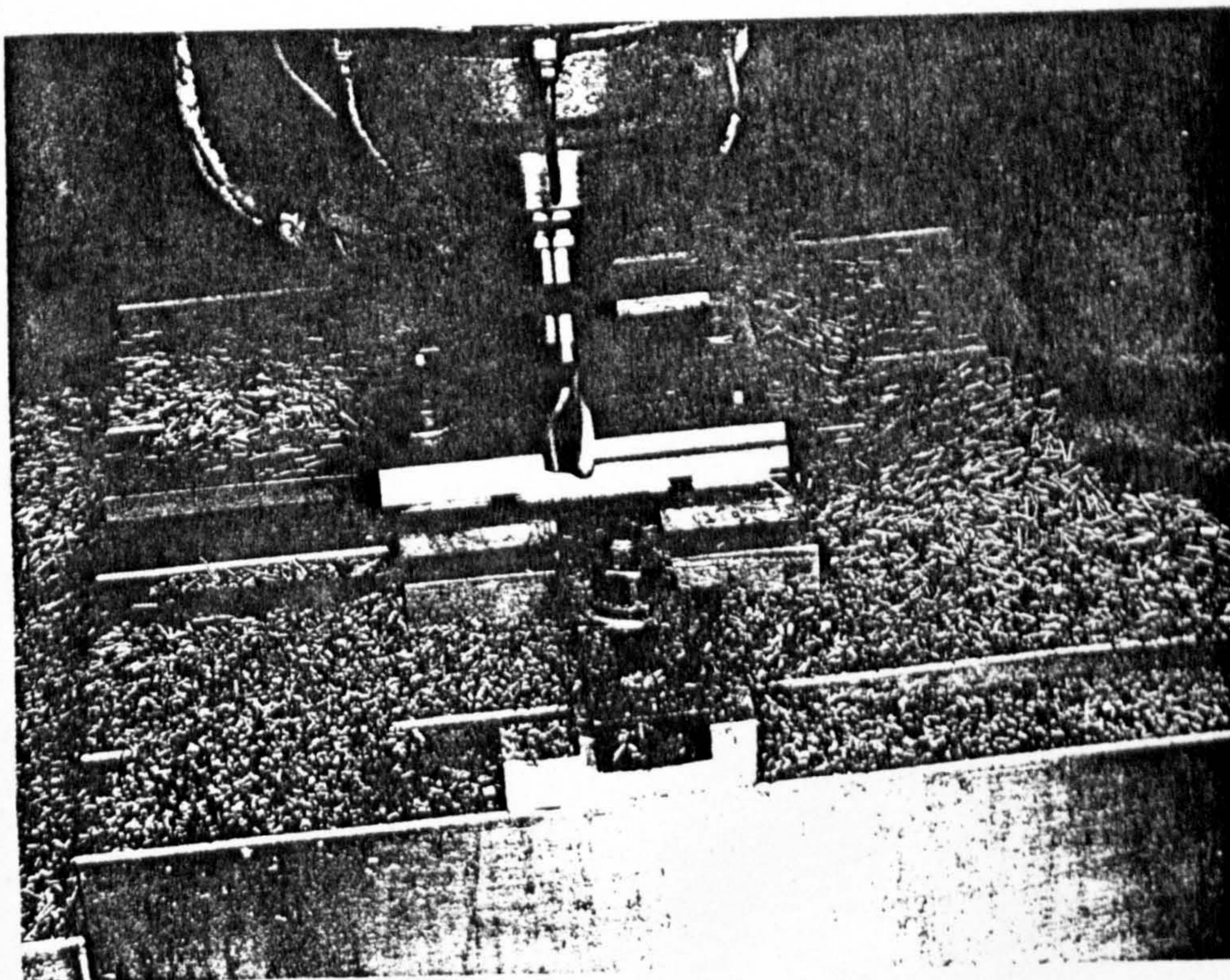


Setup 2 - Roughing the Top Face





Setup 3 - Datum Setting



Setup 3 - Profiling a Face