Aggregate process planning and manufacturing assessment for concurrent engineering

Bradley, Hugh D.

AGGREGATE PROCESS PLANNING AND MANUFACTURING ASSESSMENT FOR CONCURRENT ENGINEERING

A thesis submitted to the
University of Durham
for the degree of
Doctor of Philosophy

by

Hugh D. Bradley

School of Engineering
University of Durham

September 1997

Abstract

The introduction of concurrent engineering has led to a need to perform product development tasks with reduced information detail. Decisions taken during the early design stages will have the greatest influence on the cost of manufacture. The manufacturing requirements for alternative design options should therefore be considered at this time. Existing tools for product manufacture assessment are either too detailed, requiring the results of detailed design information, or too abstract, unable to consider small changes in design configuration. There is a need for an intermediate level of assessment which will make use of additional design detail where available, whilst allowing assessment of early designs. This thesis develops the concept of aggregate process planning as a methodology for supporting concurrent engineering.

A methodology for performing aggregate process planning of early product designs is presented. Process and resources alternatives are identified for each feature of the component and production plans are generated from these options. Alternative production plans are assessed in terms of cost, quality and production time. A computer based system (CESS, Concurrent Engineering Support System) has been developed to implement the proposed methodology. The system employs object oriented modelling techniques to represent designs, manufacturing resources and process planning knowledge. A product model suitable for the representation of component designs at varying levels of detail is presented. An aggregate process planning functionality has been developed to allow the generation of sets of alternative plans for a component in a given factory.

Manufacturing cost is calculated from the cost of processing, set-ups, transport, material and quality. Processing times are calculated using process specific methods which are based on standard cutting data. Process quality cost is estimated from a statistical analysis of historical SPC data stored for similar operations performed in the factory, where available. The aggregate process planning functionality has been tested with example component designs drawn from industry.

To my parents

and Jill

Acknowledgements

I wish to express my sincere gratitude to my supervisor, Dr. Paul Maropoulos for his guidance, encouragement and support over the period of this research. I would also like to thank Dr. Zhihui Yao for his valuable discussions and programming support and Dr. Ian Carpenter for his advice in production of this thesis. I am also grateful for the friendship, support and encouragement of the other members of the Design and Manufacturing Research Group at the University of Durham, in particular Michael Betteridge, Rob Baker, Kelvin Revere, Alima Laguda and John Darlington.

I am grateful to the E.P.S.R.C. for providing a CASE research studentship and subsequent reach grant which has enabled me to undertake this project. I am also grateful to Warner Electric (UK) Ltd for providing financial and technical support. I would like to thank in particular Mr. Jim Summerbell, who initiated the collaboration, Mr. Dave Smith for taking over with great enthusiasm, Mr. Graham Metcalfe and Mr. Mike Hemming. I am also indebted to Mr. Ian Pollard and Mr. Jonathon Heather of Caterpillar (Peterlee) Ltd, for sharing with me their industrial experience.

I would like to thank my brothers, Neil, Damian and Blaise, my sister Olivia and of course my parents for their support, encouragement and interest. Finally, I offer my thanks to Jill, who has given me encouragement and friendship throughout this endeavour.

Table of contents

List of figures

List of tables

Declaration

This thesis is the result of my own work. No part of the thesis has been submitted for any other degree in this or any other University.

Hugh Dunstan Bradley

Durham

September 1997

Notation

This section presents a comprehensive list of all the algebraic notation used within this thesis, along with a short description and the appropriate standard units, if any. For ease of reference, brief descriptions are also given at the first occurrence of any item of notation in each chapter.

a	depth of cut(mm)
α	quality factor for grinding, relates infeed to wheel width
a_a	axial depth of cut for milling (mm)
a_{amax}	maximum axial depth of cut (mm)
a_{atool}	maximum axial depth of cut for tool (mm)
a_p	effective depth of cut for a drilling process (mm)
a_r	radial depth of cut, milling (mm)
a_{rmax}	maximum radial depth of cut (mm)
a_{rtool}	maximum radial depth of cut for tool (mm)
b	manufacturing batch size
C	total cost per unit (£)
C_b	machine set-up cost (£)
C_c	coefficient in equation of Allen and Swift
C_f	coefficient in equation of Allen and Swift
C_h	piece handling set-up cost per unit (£)
C_m	material cost per unit (£)
C_{mp}	coefficient in equation of Allen and Swift
C_p	machining cost per unit (£)
C_{pk}	process capability index
C_s	coefficient in equation of Allen and Swift
C_t	coefficient in equation of Allen and Swift
C_t	transfer cost per unit between two machine tools (£)
D	cutting diameter (mm)
d	feature depth (mm)
ΔT	Difference between upper and lower tolerances (mm)
f_n	down feed for drilling (mm/rev)
f_N	down feed for drilling at maximum spindle speed
f_p	coefficient in equation of Swift *et al*
f_{pow}	down feed at maximum power (mm)
g_p	coefficient in equation of Swift *et al*
η	efficiency
h_p	coefficient in equation of Swift *et al*
i	depth of cut in grinding, (mm)
K_{cfz}	drilling resistance (N/mm^2)
K_{sm}	specific resistance to cut in turning (N/mm^2)
l	feature length (mm)
M	material removal rate (cm^3/min)
m	mean data value
M_{geom}	maximum material removal rate (cm^3/min)

m_i	current machine tool index
m_{i-1}	previous machine tool index,
m_p	coefficient in equation of Swift *et al*
N	spindle speed (rev/min),
n_a	number of axial passes in milling,
N_{max}	maximum spindle speed (rev/min),
n_r	number of radial passes,
P	Power (kW)
p	number of cutting passes,
Q	customer percieved quality (Swift)
q	production variability quality (Swift)
q_e	quality effects vector (Swift)
q_r	quality risk vector (Swift)
R	Machine cost rate (£/min)
R_c	coefficient in equation of Swift *et al*
s	table feed rate for turning (mm/rev)
σ	standard deviation of dimensions produced by a process (mm)
s_i	set-up number of current element.
s_{max}	maximum table feed rate for machine (m/min)
s_p	coefficient in equation of Swift *et al*
t	machining time (min)
t_b	machine set-up time per unit (min)
t_m	element processing time (min)
t_p	coefficient in equation of Swift *et al*
t_p	element handling time per unit (min)
v	cutting velocity (m/min)
v_c	cutting velocity for drilling (m/min)
v_w	work speed (m/min) grinding
w	feature width (mm)
w	grinding wheel width (mm)
w'	infeed for grinding (mm/rev)

Chapter One *Introduction*

1. Introduction

Chapter One

Introduction

1.1 Background

The development of new products and the improvement of existing ones is of critical importance to modern manufacturing. It is now recognised that the time taken to bring a new product idea to market has a major effect on the overall profitability of that product. This understanding has led to the development of *concurrent engineering*: "a practice of incorporating various life-cycle values into the early stages of design" (Ishii, 1993). In concurrent engineering, the aim is for all aspects of a product's development to be considered at the same time. Whilst the principles of concurrent engineering have been well established and much emphasis has been put on the management and human implications of adopting this strategy, the development of suitable tools to support it has not kept pace. In particular, there is a need for new product development tools which can assist developers performing traditionally downstream activities such as process planning. Existing tools are used too late in the development process when they can have little impact. The new tools must work early in the development process where the greatest benefits may be achieved. However to achieve this, the tools must be able to operate with the reduced amounts of information available in early design.

The lack of suitable support tools is likely to prove a serious obstacle to the implementation of concurrent engineering practice in most companies. Therefore, this work has been undertaken to develop a scheme of support for concurrent engineering through the provision of appropriate information technology tools.

1.1.1 Product Development

Product development is the process of taking an idea for a product and building that into a design, along with a means of manufacture, and eventually producing the manufactured product. Product development is increasingly recognised as a holistic activity which must address all aspects of a product's life cycle. Concerns about the impact of society on the environment are bringing the selection and planning of product disposal and materials recycling into the remit of the product development through the introduction of legislation. Overall product profitability is recognised as the critical indicator of the success or failure of the product. Most important of all perhaps is the involvement of marketing, customer and engineering disciplines in a three-way process to identify the specification for the product so that the customer is being offered a product which they want to buy.

The activities of product development are interlinked, such that a decision made during one activity causes constraints on decisions in other areas. Product development therefore requires mechanisms for the selection of the best compromises between each factor, based on an understanding of the inter-relations and the relative importance of the parameters. In traditional manufacturing companies, this goal would only be achieved by taking the product through many iterations of the tasks, until a solution was found where all the conflicts had been resolved. This resulted in a lengthy and sub-optimal process where much of the work was discarded because it did not consider constraints caused by the other disciplines in the development process. To overcome the weakness of this approach, the principle of concurrent or simultaneous engineering has been put forward. In concurrent engineering, all of the disciplines required for product development are performed in parallel, i.e. at the same time. For example, instead of waiting for the design to be finalised before determining how to make it, the production plan is developed at the same time as the design.

1.1.2 Concurrent Engineering

Concurrent engineering (CE) has been recognised as a distinct philosophy since the late 1980s, however successful companies had been using CE philosophies for many years before that (examples include Digital Equipment Corporation and Xerox's PDP system

[Hartley 1990]). CE has its historical roots in the management approaches of Japanese manufacturers, many of whom have been using CE principles, without setting out a specific terminology, since the 1970s. Hartley (1990) identifies several Japanese systems which foreshadow the philosophy of CE, in which the unifying factor is the requirement for a consensus of agreement on decisions from all members of the organisation, which leads to full commitment to the project and allows potential problems arising from a course of action to be identified from the beginning. The improvement of product development strategies follows from both a recognition of the increase in competition in world markets and a corresponding realisation of the importance of the design activity to the overall profitability of a product over its life cycle. If the design activity is too slow, then even an excellent product will not be profitable since competitors will have filled the market niche already.

Three important principles may be identified from the CE philosophy:

1. Perform activities concurrently, not sequentially, to reduce overall development time.

2. Involve representatives from all disciplines in every decision, since it is not always clear in advance where the influences will be observed.

3. Concentrate more effort and attention on early design, since it costs less to change the design at this point and the effects can be greater.

Taken together, these principles should result in shorter overall product development cycles, with fewer design corrections required, each of which costing less than in traditional product development.

1.1.3 Requirements for Concurrent Engineering

In order to implement a concurrent engineering approach, effective management of the product development process is required. In particular, CE requires both tight control over the timing of work during the project and the balancing of work effort on each task in order to minimise the waiting time. The product development process must be planned in detail, and the information requirements for each discipline must be identified. Whilst the CE methodology asks that all tasks are carried out in parallel, this is not always feasible. Sometimes it is necessary to complete certain elements of the development before further work on a particular area can continue. In order to organise the work efficiently, therefore, a detailed plan which indicates when each task will be carried out is required.

In addition to these management considerations, viable concurrent engineering practice will depend on the introduction of suitable computer aids. Designers and Engineers have become accustomed to the use of computer aided engineering tools which can dramatically improve productivity. These tools, however, have mainly been designed to operate in the traditional manufacturing environment, where product development follows a linear path. To be useful in a workplace employing concurrent engineering, these tools must be adapted. In particular, there is a need to assess the data requirements for these tools. When linear product development is carried out, each function will receive as an input the fully detailed results of the previous stage. However, in a concurrent engineering environment, the same tasks must be carried out without the benefit of full information from the preceding stages. The data from each other discipline will be supplied throughout the life of the task. Initially, only very general, conceptual information is available, whilst as the project progresses the data available will become more detailed. On the other hand, the data which is available will come from all aspects of the product development, and not just those which traditionally would have preceded the task. Thus, over the life of the product development, each function is supplied with gradually increasing data from each other function, until the fully detailed plan is completed.

1.2 Aggregate Process Planning

Maropoulos (Maropoulos 1995a) has proposed a novel methodology for process planning to suit concurrent product and process development. This approach is based on the fragmentation of the process planning function into three levels according to the detail of the task. This will result in the Aggregate, Management and Detailed (AMD) process planning architecture (Figure 1.1), where it is suggested that the process planning function will evolve into a three tiered structure, with detailed process plans being delayed until near the time of production, whilst aggregate plans will be made as early as possible to facilitate strategic decision making. The management process planning function will control the project planning of manufacture, ensure that the design and capacity constraints are satisfied and manage the manufacturing resources in the production facility.

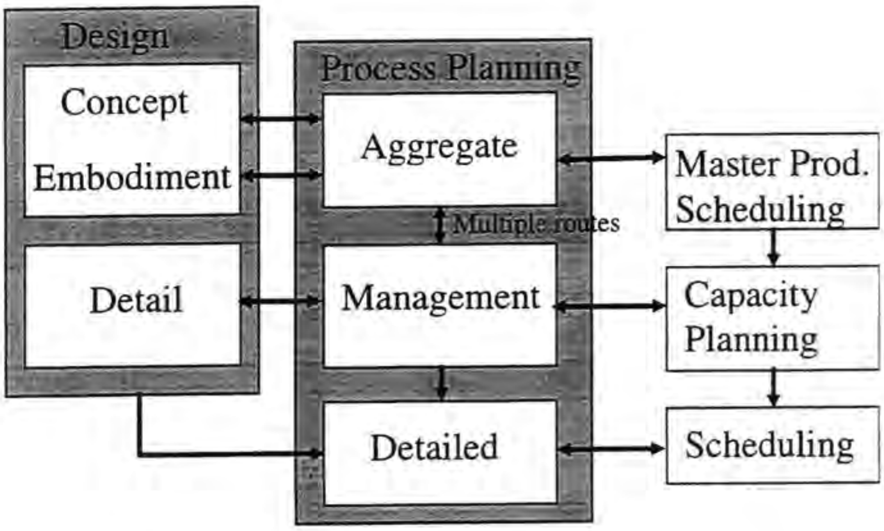


Figure 1.1: AMD Architecture for process planning

Aggregate process planning (APP) is the generation of manufacturing instructions for a given product based on a partially specified design. The aggregate production plan specifies a list of alternative manufacturing routes for each component, which include the processes to be used and the resources required. Full specification of process parameters is left to the detailed process planning stage which will be carried out when the detailed design is finalised. Aggregate process and production plans provide quantitative feedback of design manufacturability and a comparison between alternative production and processing options. Early identification of processing options allows the designs to be optimised for that process.

Aggregate process plans consist of a hierarchical set of instructions which can be mapped against a structured model of the product design. A key feature of aggregate process planning is that it identifies production alternatives and encourages the designer to explore the use of processes which might not otherwise be considered. In addition, the designer is able to receive an early breakdown of the relative costs of the product features and therefore identify the areas where additional work might result in the greatest cost savings.

1.3 Research Objectives

The objectives of this research were as follows:

- To assess the impact of concurrent engineering on computer aided engineering tools for product development in order to identify the requirements for new technology.
- To propose a new methodology for the computer support of product development which is tailored to the requirements of a concurrent engineering environment. In particular to provide support for the assessment of manufacturability in the early stages of product design.
- To develop and implement a prototype computer based system which provides the functionality identified. It is expected that a concurrent engineering tool should provide integration between design and production knowledge. In particular, the ability to identify and evaluate alternative options is critical to early product development.
- To evaluate this prototype system by through testing with industrial designs in order to allow comparison with existing engineering methods: The system will be judged on criteria of ease of use, impact on design time, accuracy of data and the ability to cover a wide range design configurations and processing options.

The prototype computer system is to serve as a test-bed for the ideas which are proposed in this thesis; it is not intended for commercial use.

1.4 Outline of Thesis

Chapter 2 presents a review of research in the fields of concurrent engineering and product development support. A general overview of the computer system which has been developed is given in Chapter 3, showing how the system works as a whole. The next three chapters deal with the individual models which have been developed for the system: In Chapter 4, the aggregate product model which forms the input to the system is detailed. Chapter 5 details the process models which are used with the process planning assessment and describes the cost and quality calculation methods which are used. Chapter 6 discusses the way in which resource information such as factories, machine tools and labour are modelled. Chapter 7 details the implementation of aggregate process planning within the system. The process planning functionality and the route model which stores process plans are described. Chapter 8 presents the results of the testing of the system with example data, including examples from industry. A summary of the work, including suggestions for future work and conclusions, is presented in Chapter 9.

1.5 Related publications

This thesis presents the author's own work except for appropriately acknowledged related work. Earlier work in progress and software developments have also been documented in internal reports of the University of Durham, technical articles and refereed papers. It is the intention of the author that the work described in this thesis will be published subsequently in further papers. Previously published papers relating to early versions of this work include:

- Maropoulos, PG, and Bradley, HD, An object oriented process modeller for concurrent engineering environments, *Proc. 1st Joint conference of simulation societies, Zurich*, 1994.

- Bradley, HD, and Maropoulos, PG, A concurrent engineering support system for the assessment of manufacturing options at early design stages, *Proc. 31st Intl. MATADOR conference, Manchester*, 1995, 485-492.

- Bradley, HD, and Maropoulos, PG, A relation based product model for computer supported early design, *Proc. 13th CAPE conference, Warsaw*, 1997, 57-64.

- Maropoulos, PG, Bradley, HD, and Yao, Z, Capable: An aggregate process planning tool-kit for integrated product development, *Proc. 13th CAPE conference, Warsaw*, 1997, 49-56.

- Bradley, HD, and Maropoulos, PG, Aggregate process planning: A methodology for supporting concurrent engineering, *Proc. 32nd Intl. MATADOR conference, Manchester*, 1997, 513-518.

Chapter Two

Literature Review

2.1 Introduction

Having established that the successful integration of concurrent engineering methodology into the product development process of a manufacturing company will result in a great improvement in productivity and performance, the task is then to determine the requirements to achieve this purpose. This chapter presents a review of the state of manufacturing systems technology in the light of the introduction of concurrent engineering. Furthermore, the requirements for improvements and alterations in the technology are identified and research efforts in this area are reviewed. In particular, this survey covers the methodologies and systems which have been proposed for the support of concurrent engineering. It is suggested that the modelling of product designs and manufacturing processes together with the integration and management of data are of particular importance in the pursuit of concurrent engineering manufacturing systems and these areas are addressed in detail at the end of the chapter.

2.2 Product development

Product development is the principle business of a large proportion of manufacturing companies. In order to understand the impact of concurrent engineering on product development, it is necessary to review the task of product development and identify each of its elements. Numerous studies have attempted to systematise product development, in particular to identify distinct stages which must be undertaken. BS7000 (standard for engineering management), identifies eight stages within the life cycle of a product: Product planning, feasibility study, design, development, production, distribution, operation (use) and disposal (Figure 2.1). The standard uses the traditional model for the design process, splitting design into four stages of conceptual, embodiment (also known as configuration design), detail and design for manufacture (or production planning). Pahl and Beitz (1984) similarly identify four stages of design: clarification of the task, conceptual design, embodiment design and detail design. At each stage of this process the potential number of options for the developer will be greatly constrained by the previous choices. In the ideal case, CE allows a greater range of options by considering the later stages of the life-cycle (e.g. production and disposal) before committing to detailed design. In addition, whereas the traditional approach leads to backtracking along the development path when unfeasible suggestions are only recognised after further stages, with CE designs should theoretically be right first time.

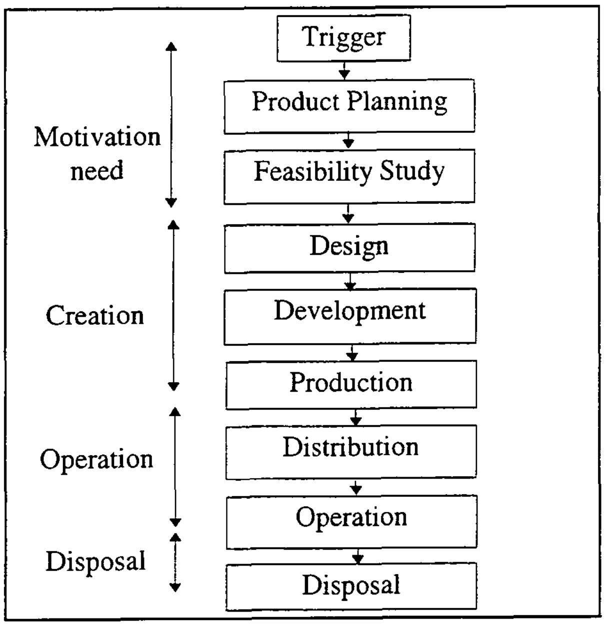


Figure 2.1: Traditional product development (BS7000)

Concurrent engineering requires a new paradigm for management of product development where the priorities of management have changed: the emphasis is now on constraints on the scheduling of activities (precedence relationships, manpower availability), provision of the required information at the right time and performing activities with incomplete data.

Andreason and Gudnason (1992) stress the importance of planned product development. They review the various managerial approaches to improving the product development cycle and then discuss the importance of integration of the various activities. The concept of *dispositions* (Andreason and Oleson, 1990), is introduced, where a decision in one area creates a disposition which constrains another area, the "victim". The importance of feedback from the victim area is established. Kunz *et al* (1996) discuss the way in which concurrent engineering has been implemented in companies; they maintain that CE implementation often only links product and process, instead of including the design of the manufacturing facility and organisation. A schema for the integration of organisational and facility design with product and process development is presented.

Sohlenius (1992) presents an overview of the early impact of CE, stressing the need for education and for good team-working skills for its success. Krause and Ochs (1992) similarly identify the need for team work in CE. The blackboard architecture for problem solving is explored, in which multiple problem solvers (or agents) interact with a single representation of the solution. The possibility of network based working and computer based agents is discussed. Control of the blackboard is seen as the primary difficulty in applying this architecture, requiring a CE-specific logic to be applied to the system.

2.3 Manufacturing technology

Until the widespread recognition of the value of concurrent engineering at the beginning of the 1990s, manufacturing technology was geared up to support the "traditional" product development process. It is natural, therefore, that the introduction of a radical change in the strategy of product development should require a complete review of the associated technology and the introduction of new or modified support tools.

2.3.1 Management of manufacturing

In traditional manufacturing companies, the engineers involved in product development have been organised into departmental groupings based on engineering discipline. In addition, many of those engineers actually involved with the process of developing a product will not have been considered as such because they belonged to other functional groups within the enterprise, such as production management. In CE, the importance of the involvement of participants from all of the disciplines affecting product development throughout the life cycle of a product is acknowledged (Hartley, 1990). When possible this product development team should include representatives from the customer of the product and from the suppliers of materials and sub-components.

Research in the field of management for manufacturing has therefore identified a number of areas. In particular, the logistical difficulties of multi-functional teams have been addressed through the concepts of *virtual team working* and network communications (Toye et al, 1994). The task of *engineering data management* (EDM) has been recognised as an important factor in managing product development of any form, but is particularly important in a concurrent engineering environment. In the field of artificial intelligence important work has been carried out to investigate the requirements of teams to negotiate and reach compromises. This work has applications for both human and computer based multi-agent co-operation (see section 2.5).

2.3.2 Project planning

The introduction of concurrent engineering has focused research and management attention on the task of project planning. Time compression is one of the two key principles of concurrent engineering (the other being to get decisions right first time) and this can only be achieved through detailed planning of the tasks of product development. In an ideal situation, truly concurrent engineering would be possible, with all aspects of the product development occurring simultaneously. In reality, however, there will always be some aspects of a design which must be performed before another decision can be made. Project planning can be considered a tool which enables the real situation to come as close as possible to the ideal.

Eppinger *et al* (1994), suggest a methodology for planning large scale development projects by using binary *design structure matrices* (DSMs) to represent the precedence relationships between development activities. Several versions of DSMs are discussed, with enhancements made to allow the technique to represent additional relationships between activities such as feedback, control and addition. The aim of this research was to identify methods to cluster activities and improve the efficiency of the development process. Future developments of the work are identified as the introduction of parameter level models for detailed design. This technique is similar to that employed by Kusiak and Wang (1993a, 1993b), who suggest decomposition as a means of enhancing concurrency in design. They describe a methodology for decomposition of the design process, based on analysis of the relationships between task and design parameters, in order to optimise concurrency. Decomposition is performed by clustering the task-parameter incidence matrix to decompose it into several sections which may be performed concurrently. This approach applies a generic and formal methodology to planning in the design phase which can enhance concurrency. In order to apply this system, however, a detailed understanding of the design process is required, including the identification of all design parameters. This means the approach is most suitable for redesign.

2.3.3 Computer Aided Engineering

Computer aided engineering (CAE) tools owe their development to attempts to provide support for individual product development activities which were perceived as bottlenecks in the development cycle. Initial CAE tools were developed to help engineers with specific problems such as gear selection and bearing design, in the manner of automated catalogues. Other CAE tools were introduced to apply the power of computers to the solution of complex analysis problems, such as structural mechanics, e.g. *PAFEC* (Woodford *et al*, 1992) and computational fluid dynamics, e.g. *PHOENICS* (1992), typically using finite element analysis (FEA) methods. Later FEA tools provide integrated systems of analysis of multiple domains, e.g. *Strand 6* (Clarke, 1997). Computer aided draughting was introduced to reduce draughting times, particularly in re-design. This later developed into the greatly enhanced form of computer aided design (CAD) through the introduction of three dimensional part representations. Automated machine tools and robots required programming tools, which contributed to the introduction of computer aided process planning (CAPP).

Shop floor management and facilities design benefited from the development of simulation as a tool for investigating production flow on the shop floor.

With the exception of CAD, the problem with CAE tools is that the time required to input the data necessary to achieve useful results is equivalent to the time needed to solve the problem. Furthermore, these systems are not connected, instead forming "islands of automation". Once CAE is adopted, therefore, further reductions in development time can be achieved mainly by reducing data input times. The best way to achieve this is to input data only once and have it passed from one CAE tool to another automatically. This lead to the concept of computer integrated manufacturing (CIM). A review of CIM technology is presented by Lu (1990). It is generally agreed that the main enabling technology for CIM is the development of product data interchange methods and compatible CAE tools, subjects which are discussed in more detail below.

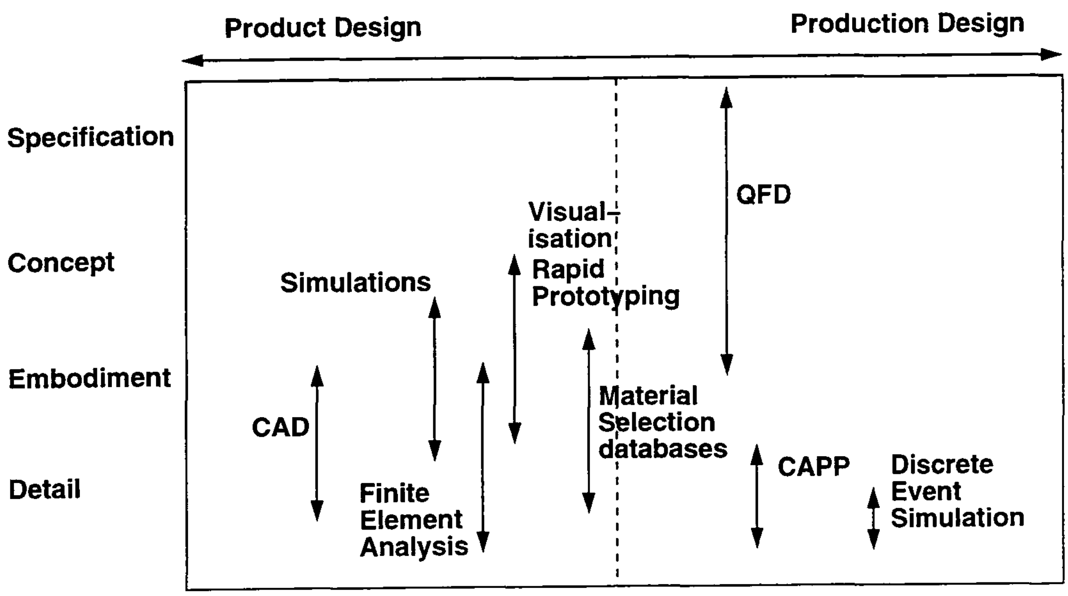


Figure 2.2: The influence of CAE tools during product development

Concurrent engineering requires the re-evaluation of CIM strategy and CAE tools. Merely creating a fully integrated manufacturing environment where all development activities are supported by CAE tools linked by a data exchange mechanism will not result in CE application but rather an automating of traditional "over the wall" design. CE environments require new CAE tools which are multi-disciplinary so that the implications of each decision and the full constraints are considered. Examining the range of CAE tools currently available (Figure 2.2) it can be seen that there is a strong bias towards the detailed design phase. CE requires that more emphasis is placed on the conceptual and embodiment phases and therefore new CAE tools must be developed. The characteristics required for CAE tools for concurrent engineering will be:

- multi-disciplinary,
- open to access to all project members,
- able to operate on variable levels of detail,
- able to operate with incomplete data

2.3.4 Design

Research into improving the design phase can be divided into two main categories: Design theory seeks to understand the process of design and to develop new methodologies which can be applied; design for X (DFx) research is a field which seeks to develop specific methods for optimising the design for a particular objective such as assembly, (design for assembly, DFA), where the designer applies rules which are intended to reduce assembly difficulty and therefore cost. Several researchers have attempted to develop a generic model for design.

2.3.4.1 Design Theory

Peters *et al* (1990) define engineering design as "*a creative operation of products, production processes or more generally of humanity serving systems using available materials, products and processes, aiming to achieve a specific function, responding to the demand of the customer and compromising between conflicting constraints such as cost, delivery, delay, feasibility and maintainability*". This definition identifies the key problems for design:

- The need to be creative.
- Analysis of ideas.
- The need to map functions to the requirements of the customer.
- The need to map systems to the functions.
- The need to satisfy constraints.

Researchers have attempted to meet these needs in a variety of manners: Karni (Arciszewski and Karni, 1997) has developed *ideation* as a methodology for improving creativity at the conceptual design stage; Quality function deployment (QFD) (Akao, 1990) is a successful tool for integrating the customer's requirements into the design process. Several researchers (see Thornton, 1993) have viewed design as a constraint satisfaction process in order to develop new design methodologies. This review concentrates on design research which is concerned with the improvement of design in concurrent engineering terms.

Andreason (1991a) presents an in-depth analysis of design methodology. It is argued that design activity may be measured according to six universal values: Quality, Cost, Efficiency, Flexibility, Risk and Time. These criteria can be used to evaluate the performance of alternative design systems. The need for detailed, structure models of product development is seen as critical in order to determine how to implement improvement methods such as "design for X" methods. The importance of a clearly understood, concurrent design process which can be managed is paramount.

2.3.4.2 Design for X

The traditional design procedure of specification, conceptual, embodiment and detailed design recognised no requirement for the consideration of manufacturability during the design process. The production planning activity was to follow after the optimum functional design had been determined. Whilst a good design engineer would always have the eventual manufacture of the product in mind, there was no formal process involved and no structured method of ensuring that the design was capable of being manufactured. Designers do not necessarily have experience and knowledge of the manufacturing processes which are available to realise their designs.

Design for X (DFx) is a generic term for a set of methodologies which have been developed to improve the link between design and downstream development activities. The first DFx methodology was DFA, developed by Boothroyd and Dewhurst from 1977 and first published as a handbook in 1980 (Boothroyd and Dewhurst, 1987). The DFA methodology uses four steps:

(i) Formal method questioning whether every part is necessary to calculate *theoretical minimum number* of parts,

(ii) Calculation of estimated assembly time,

(iii) Derived DFA index based on comparison with *theoretical minimum index* (calculated as time to assemble minimum number of parts if they are easy to assemble).

(iv) Identification of difficulties.

Boothroyd and Alting (1992) present a comprehensive review of work related to DFA systems and identify the need for Design for Disassembly (DFD): With the increasing concern for environmental issues, regulations are anticipated that will make producers responsible for the disposal of their products. This will require that the cost of disposal be minimised. The DFD methodology is a rule based system which aim to reduce costs by eliminating hard to disassemble and environmentally expensive parts.

Since DFA aims to reduce the number of parts solely through the minimisation of assembly cost, the individual cost of components may increase. To manage this problem, the concept of design for manufacture (DFM) was introduced (Andreason, 1991b). A number of different approaches to DFM have been suggested by researchers. Many DFM methods involve a sets of guidelines and checklists relating design features to particular processes in order to generate a "good design" which is suitable for that process. Other research has attempted to determine quantitative measures of manufacturability for design, for example, Allen and Swift (1990) proposed a model for manufacturing cost prediction based on a comparison of the component with an ideal design for each specific process. They define a *relative cost coefficient*, R_c thus:

$$R_c = C_{mp} \,.\, C_c \,.\, C_s \,.\, (\max.(C_f,\ C_t))$$

Where: C_{mp} = material-process suitability coefficient,

C_c = geometry suitability coefficient,

C_s = section thickness suitability coefficient,

C_f = tolerance suitability coefficient,

C_t = surface finish suitability coefficient.

In the ideal case, the value of each coefficient is unity, but as the design moves away from the ideal, one or more coefficient will increase. Other manufacturing technologies can be modified for use in a DFM context, including group technology, value engineering, QFD and failure mode and effect analysis (FMEA) (Andreason, 1991b).

Quality oriented DFM methods (or design for quality, DFQ) have been developed to reduce quality costs and improve process capabilities. Krause *et al* (1993b) have developed a methodology for the integration of quality tools during design in order to achieve DFQ. In particular, a feedback architecture is proposed using a quality information system (QIS) to support QFD and FMEA methods. Johnston and Burrows (1995) describe a computer implementation of QFD using house-of-quality charts implemented on a windows system. This system suffers from a lack of integration with CAE and product models. Swift *et al* [(Swift and Allen, 1994), (Batchelor and Swift, 1996)] propose a methodology based on risk assessment in order to develop robust designs. In this approach, two categories of quality are identified: Q, customer perceived quality requirements and q, production variability quality. The former term is a vector used to show the relative importance the customer places on particular product attributes. The latter term is used to define the quality risk, vector q_r, which is expressed as the relative risk compared with the ideal situation. Two models are used, one for components and one for assemblies:

- Components: $q_r = q_m = m_p \cdot g_p \cdot t_p \cdot s_p$
- Assemblies: $q_r = q_a = h_p \cdot f_p$

where: m_p = material to process risk

g_p = component geometry to process risk

t_p = tolerance to process risk

s_p = surface finish to process risk

h_p = handling process risk

f_p = fitting process risk

The overall component or assembly quality risk is determined by using QFD, combining both Q and q_r with an effects vector q_e.

Andreason and Oleson (1990) introduce the concept of dispositional mechanism as a generic pattern for DFx methods. A dispositional mechanism is a description of the way in which an upstream development decision affects the decision environment of a downstream activity, called the victim activity. Any development activity determines the task for a later activity. The disposition is the data which determines the conditions under which the victim task can be solved. Dispositional mechanisms are modelled using targets in the form of rules, standards and data. The aim of these targets is to ensure that the upstream decision leaves the victim activity with a valid solution space.

Additional examples of DFx methodologies include design for serviceability (Gershenson and Ishii, 1993) design for the environment (Boothroyd and Alting, 1992) and design for reliability (Birolini, 1993). It can be noted that several of the fields overlap, particularly design for disassembly (DFD), which is part design for serviceability and part design for the environment. Whilst DFx methodologies can prove valuable in improving designs, they do not provide an integrated solution to the requirements of product development. Each DFx system tends to give priority to one aspect of the product development, whereas concurrent engineering emphasises the need to consider all aspects together.

2.3.4.3 Computer aided design

Traditionally engineers have used technical drawings to record and communicate design information. The first computer aided design (CAD) systems were essentially drafting systems which allowed the generation of two dimensional (2-D) drawings on a computer. The principal advantage of these CAD systems was that they allow the re-use of existing designs for new products through modification of drawings. A major disadvantage of all 2-D CAD systems lies in the difficulty of design visualisation. The 2-D representation does not provide the geometrical information to allow analysis of the design.

The first attempts to model designs in three dimensions were wire frame modellers. In a wire frame model, the solid objects are described as a collection of edges and vertices. Whilst wire frame models are an improvement over 2-D models, they have limitations in the geometry which can be represented and present difficulties in interpretation of the model for analysis. These were followed by surface modellers, which represent the part by using mathematical descriptions of the component surfaces. Pure surface modellers are difficult to use, particularly for adding and manipulating information.

The current state-of-the-art in CAD geometry is the use of solid modelling systems. These are described in detail later in this chapter (section 2.4.1). Current research into CAD is now divided into a number of areas. In addition to those discussed elsewhere, is the concept of Intelligent CAD (ICAD) (Yoshikawa, 1993). In an ICAD system, the geometric modelling and representational capability is combined with the capturing of design knowledge. According to Yoshikawa, there are two common means of capturing design knowledge using a KBS: either a very large number of *shallow* rules are stored in the hope of covering all eventualities, or a model-based reasoning system is devised that has a deeper understanding of the situations so that fewer rules may be applied to a wider set of circumstances: A hybrid approach is considered favourable. In this situation, there is a need for a systematisation of the knowledge in order to manage it. Systematisation is the conversion of recognised and tacit knowledge to codified computable knowledge. This technology has been applied in building a demonstration system.

2.3.5 Process planning

Process planning is recognised as one of the most important activities in product development. The introduction of automated production machinery has resulted in an increase in the detail which is required for production plans: CNC machines require complete programming whereas for manually controlled machine tools the fine details could be left to the machinist. It is not surprising, therefore, that the field of computer aided process planning has been one of the most thriving research areas since CNC machines were introduced. Many comprehensive reviews of computer aided process planning (CAPP) research have been published in recent years [(Elmaraghy, 1993), (Ajmal and Zhang, 1994), (Maropoulos, 1995b, 1995c), (Carpenter, 1996)] and therefore it would be inappropriate to attempt to give a full review of CAPP in this thesis. Instead, a summary of the main approaches to process planning is given and an interpretation of future trends is made.

Process planning involves a number of elements, particularly: selection of process, selection of tools and equipment, selection of process parameters, generation of machine instructions. Elmaraghy (1993) suggests a classification of process planning technology into four levels, according to the detail involved:

- *Generic* (or conceptual) process planning is concerned with the selection of suitable production technology for the part and with providing rapid feedback to the designer so that designs may be optimised for the process. This planning is similar to the concept of aggregate process planning (Maropoulos, 1995c). An example of this method was developed by Esawi (1994).

- *Macro* planning is concerned with routing and sequencing. Such systems are characteristically multi-domain, i.e. able to consider several process technologies. An example of a macro planning system is COMPLAN (ESPRIT 6805, 1995).

- *Detailed* process planning systems are typically specific to a single domain, or able to consider only one at once. These systems are concerned with selection of tools and resources and sequencing of operations. Example detailed CAPP systems include TECHTURN (Davies, 1991), PART (van Houten *et al*).

- *Micro* planning is concerned with the optimisation of a single process operation. These are nearly always single domain systems. Examples include OPTIMUM (Carpenter, 1996).

This view of CAPP neglects the importance of time considerations; for relevance to CE research, it is the timing of process planning is an important factor. CE environments require that different levels of process planning are performed throughout the product development purpose (see Figure 1.1), ranging from aggregate planning during conceptual and embodiment stages, to detailed planning when the designs are finalised. If appropriate CAPP tools are available at each stage of product development the quality of design will be enhanced. An important division in CAPP research systems can be made according to the way in which the plans are generated:

- *Variant* planning systems produce process plans by searching historical databases for similar products and modifying the plans for the appropriate new component. These were the first CAPP systems, requiring the least complex computer systems. Variant planning allows the creation of more complex plans since they are not restricted to a product data model. Variant planning has been the most widely implemented method in industry.

- *Semi-generative* and *generative* systems are equipped with the ability to make process plans from a set of rules based on first principles. These systems require more information about both processes and parts. Early examples using GT coding schemes were replaced by systems using part description languages. The concept of a part description language has now been superseded by the concept of the product model. Generative process planning has the advantage of flexibility, permitting alternative production methods to be considered.

Practical CAPP systems can be divided into two groups according to the approach which is taken to automation. On the one hand, *automated* systems aim to perform complete process planning without user intervention. They are faced with the problems of limiting the search space which is often too large to analyse in a reasonable time. On the other hand, *interactive* systems perform the planning process in steps, frequently requiring user input to confirm or refine the plans. These systems must balance the need for user intervention with the ability to provide rapid results.

From a CE point of view, the links between the functions of process planning and other product development disciplines are important: Process selection provides a feedback to embodiment design, so that designs may be optimised for the production method; The production routing draws on facility layout data and also can provide feedback into facility design such as cell clustering; process plan data can be used in simulations to balance production lines. The selection of process parameters is the focus of many detailed CAPP systems, where the aim is to optimise the production rate or cost. Parameter selection is less important at aggregate level, however, since the design details will likely be altered in the final design. At this stage then, it is more important to provide indications of costs, both for comparison between process alternatives and for identification of design improvements.

There are two main weaknesses in conventional CAPP research: Firstly, there is the need to make CAPP systems robust enough to operate in the industrial environment. In most manufacturing companies the production environment is subject to frequent change due to the introduction of new products, disruptions due to maintenance, staff absence, quality problems etc. This often leads to production plans being changed on the shop floor to meet demand. However, such *ad hoc* planning is seldom efficient and can cause even greater problems to other products. The second weakness is that, from a concurrent engineering viewpoint, process planning should be integrated with the design process so that design decisions are made using process planning criteria as well as functional criteria. Detailed process planning systems are not suitable for this integration, since they cannot operate without the detailed product designs, which are not available until the end of design and they are restricted to particular processes. There is a need for real-time evaluation of process planning implications of each design decision. Since CAPP technology is not able to provide this, the concept of DFM has attempted to provide an alternative, assessing designs without actually producing process plans.

The particular requirements for a CE oriented CAPP system can be summarised as:

1. The generation of process plans from incomplete data

2. The identification and analysis of alternative production processes

3. An awareness of the impact of each alternative process plan on the production environment

A number of CE oriented CAPP systems have been developed, many of which form part of CE systems described later (section 2.6). The ESPRIT project COMPLAN (ESPRIT 6805, 1995) implements a new process planning architecture based on the concept of non-linear process plans (NLPP) in a prototype system. Process plans are represented as a net of alternatives instead of the conventional linear sequence. This approach recognises that process plans will often change at the time of production because of scheduling problems and seeks to anticipate this and plan alternative routing options. NLPPs require greater processing to link scheduling to the normal process planning activity. The COMPLAN project uses the commercial AVOPLAN CAPP system to provide automated process planning of detailed designs.

Herman *et al* (1993) proposed a flexible and opportunistic style of process planning, implemented in the system XTURN. This system is a flexible decision support environment, based on experience-based heuristics and multi-directional process models. Process plans are generated interactively: as the designer adds information (and therefore constraints), the system identifies feasible tooling and checks the manufacturability. XTURN is notable for its attempt to implement least-commitment planning, where incomplete data can be used to create sets of alternative plans depending on the future decisions. Also, the process planning implementation is concurrent with design instead of a succeeding activity. There are disadvantages to the system, however, particularly in the restriction to a single process domain. Few parts are purely axi-symmetric and optimal plans for parts involving multiple processes should consider all processes together.

2.3.6 Facility design

The decision whether to re-design the shop floor for a new product, including possible capital equipment purchases, depends on volume, production strategy and capacity. For true CE implementation, the design of the facility should be closely integrated with the rest of product development. Various computer tools are available to support facility design, including discrete event simulation to analyse flow patterns and identify bottlenecks (Chan et al, 1995), clustering algorithms for machine cell creation (Kandillar, 1994) and virtual reality mock-ups to check access and ergonomics (Qing and Ming, 1997).

2.3.7 Prototyping, design testing and production proving

The creation of prototypes and their analysis has been a necessary part of the product development process in the past, as the only reliable means of testing the feasibility of a design or a production process plan. Today, however, considerable research and development has been undertaken in order to reduce the requirements for prototypes. In particular, the development of the practice of rapid prototyping has been advanced to allow the visualisation and understanding of the proposed design even at a conceptual stage. Rapid prototyping (Kruth, 1992) is concerned with generating the shape of the object, without restricting itself to high quantity production facilities. Modern CAD systems have now developed to the extent whereby prototypes are often used only as a final check, since the capability for the generation of photo-realistic images on computers allows the visualisation of the design concept. Furthermore, the development of computer analysis tools has provided the opportunity to perform accurate assessments of product function within computer simulations, instead of testing a physical prototype (Cartwright, 1997). Examples include FEA and dynamics simulations, offered in commercial CAD systems such as Pro/Engineer, Euclid and CATIA.

Whilst it is still necessary in most cases to have a trial run of the production on the real equipment, in order to ensure high quality, there are methods for ensuring that production goes smoothly from the start. These include computer tools to check production, specifically discrete event simulations and tool path checking algorithms for use with CNC programs.

2.4 Product Modelling

Product modelling is at the centre of current research into integrated product development. A product model is a means of storing and representing data about the product. Historically product models have been designed specifically for a particular CAE element, to perform a particular task. Examples of early product models include 2D CAD representations, finite element mesh representations and dynamic system simulations. Each of these models performs adequately for a specific task, yet it is nearly impossible to translate one model automatically into another: Each model stores and represents only a subset of the total product data, according to the task at hand. In addition, the different nature of the engineering disciplines involved lead to a fundamentally different approach to modelling the product (Salomons *et al*, 1992). This acts to prevent the transfer of data from one system to another. The importance of a properly structured and powerful product model can be seen from the requirements to improve the integration between CAE elements. It has been suggested (Spur *et al*, 1986) that data flow in product development can be classified as either geometry oriented data or administration oriented data. In order to manage the integration of separate software systems it is important that these data flows be incorporated into the same model.

A comprehensive attempt to define a specification for product modelling is presented by Krause *et al* (1993a). They define a product model as "the logical accumulation of all relevant information concerning a given product during the product life-cycle". It is further suggested that the model should be stored digitally and be equipped with access and manipulation algorithms. A methodology for the design of product models for specific manufacturing systems is set out using the concept of *process chains*, which represent the set of technical and management functions required to develop products from beginning to end. Peters *et al* (1990) give a historical perspective of product modelling. The requirements of a product model can be summarised [(Krause *et al*, 1993a) (van der Net *et al* 1996)] as:

- Create a consistent product description for all stages in design and manufacturing
- Present the actual model data
- Capture and record the design intent
- Facilitate product documentation
- Offer decision alternatives
- Ensure manufacturability whilst designing

The model can help prevent unnecessary iterations in the design process in various ways: By maintaining alternative decisions it provides protection from downstream uncertainties; Manufacturability checks can immediately identify some impossible or undesirable designs.

Arai and Iwata (1992) discuss the specific requirements for product models in the conceptual design stage. In particular the need to integrate functional modelling with geometry modelling is stressed. This approach is supported by other researchers [(Salomons *et al*, 1992), (An *et al*, 1995), (Meerkamm, 1993)]. The authors state that a conceptual product model should support representation of (i) functional requirement, (ii) design specification and (iii) rough structure of design solution. In order to link functional and geometric modelling, the representation of the designer's intent is critical. A structured "Design Process Description Language" (DPDL) is proposed as a means of standardising the design intent of a particular action. This is an attempt to devise a language which may be processed automatically or by humans to pass on the design intent. The DPDL is expressed in the format

[ ] --> [ ] or [ ] --> { }

where each [ ] represents an intention, --> represents the thought flow and { } an action, such as a modelling operation. Intentions are described with labels and a set vocabulary.

Another approach to the capturing of design intent is presented by van der Net *et al* (1996), using the concept of manufacturable design transformations. In this modelling system, the designer is restricted to a pre-determined set of manufacturable geometric transformations that are characterised by an operator and an associated design object, which is represented in the resulting model as a reference element linking features together according to either topology, tolerances or assembly relations. The advantage of this approach is that design manufacturability is ensured and downstream users of the model can see the relationships intended between features. However, this scheme does not capture functional design intent at this stage.

In summary, the requirements for product models go beyond merely representing the product from the point of view of one engineering discipline. The product model should provide an integrated data set which maintains all product data from initial concept to disposal. This means that the product model must be capable of changing with the evolution of the product and supplying data in formats suitable for all engineering disciplines.

The product models available from commercial vendors do not currently meet these requirements. In general, product models are geometrical models based on CAD systems. A number of software tools are available which claim to offer an integrated CAE environment, based on a core CAD system (Pro/Engineer from PTC, Euclid from Matra Datavision, Catia from Dassault/IBM). Each of these uses a primarily geometrical model to represent the product although additional data may be stored in some cases. In order to provide integrated finite element analysis, the most advanced systems allow the automatic generation of meshes from the product geometry model. Similarly, there are software kits available which can provide dynamics analysis of CAD solid models. However, none of these models provides a suitable solution to the representation of design intent and most are inadequate for the demands of analysis such as automated process planning tasks.

Current research into product models has concentrated on enhancing geometric product models, either to produce an integrated product model suitable for all product development domains or to tailor the model for use in a particular domain.

2.4.1 Geometric models

A product model requires a geometric model in order to represent and allow reasoning on the design geometry. The current state of the art in geometric modelling is the solid model. There are two main approaches to solid modelling, boundary representation (B-Rep) and constructive solid geometry (CSG). In addition, many systems now combine these approaches into a hybrid B-Rep/CSG scheme.

In boundary representation, the part is described by a face-edge-vertex graph which explicitly represents all geometric and topological information. The disadvantage of B-Rep systems is that models are difficult to construct and they are poor at capturing the design intent. Constructive solid geometry defines the part using a set of volumetric primitives such as cylinders and cuboids, which can be combined using Boolean operators. CSG representations are concise, simple and powerful and easy to edit. The main drawback to CSG representations is the lack of explicit representation of the lower-level entities of the part, such as lines, points and surfaces. Hybrid solid models seek to combine the advantages of both solid modelling approaches. CSG representation is used for the macroscopic representation of geometry, whilst lower level entities are represented through the modelling of each CSG primitive in B-Rep format.

2.4.2 Feature Technology

The main method of enhancing geometric models has been through the introduction of the *feature* concept. A feature is a subset of product model data which can be used for reasoning about the product. A large body of research work has been generated on feature based product models [(Case and Gao, 1993), (Salomons *et al*, 1992)]. Many researchers have attempted to define the term features, but there is much disagreement over the use of the term. What is commonly accepted is that the term features is defined differently according to the point of view of the research. Definitions range from the general: "a feature is a region of interest on the surface of a part" (Pratt and Wilson, 1985) to more specifically related to a particular domain, e.g. Henderson *et al* (1990) adopt a definition oriented towards representation and recognition of features: "features are defined as geometric and topological patterns of interest in a part model and which represent higher level entities useful for analysis". Van't Erve (1988) defines features for process planning: "a distinctive characteristic part of a workpiece, defining a geometrical shape, which is either specific for a machining process or can be used for fixturing and/or measuring purposes". Lenau and Mu (1993) suggest two complementary definitions of features: "Information sets that refer to aspects of form or other attributes of a part" and "a group of geometric entities that together have some higher-level meaning". The first definition is more general, whilst the second limits the term to geometric entities. Fu *et al* (1993) define features thus: "A feature is an abstraction of a set of geometric constraints and can be associated with a meaningful context". Example contexts are either manufacturing or functional.

It is clear that the term features can be applied to a wide range of different entities in product modelling. Some researchers use features to represent information relating to just a single domain, whilst others attempt to define generic features which can be used in all engineering domains. Many different types of features are identifiable, the three main types being:

- Design or Functional features
- Manufacturing features
- Purely geometric features (form, shape or geometric features)

Alternatively, features may be viewed as either *geometrical* features, representing only the form of the product or *application* features, which are suitable for use in tasks such as engineering analysis, process planning and inspection. The same geometry will be interpreted into a different set of features depending on which classification is being used. There are two ways in which this problem is resolved by current feature research: (i) permit different applications to have different feature sets and write methods to map features between them, or (ii) adopt a unified feature representation across all applications.

Feature related research can be divided into two main fields: the representation and data structures of features and the means of obtaining the feature data to create the model. The former may be viewed as the development of feature taxonomies, whilst in the latter case, two approaches dominate: design by features and feature recognition.

2.4.2.1 Feature taxonomies

A feature taxonomy is a hierarchical classification of different features into classes and subclasses according to common properties. A feature taxonomy is central to the development of a feature based product model and many researchers have developed feature taxonomies. The failure of a standard feature taxonomy to emerge can be explained by Case's assertion that "the way of classifying features is highly dependent on feature representation methodologies and strategies for the eventual use of the feature data." (Case and Gao, 1993).

Butterfield *et al* (1985) adopted three main classes of form features: sheet features, rotational and non-rotational. Each of these classes was further divided: sheet features as either flat or formed, rotational as either concentric or non-concentric and non-rotational as either depressions, protrusions or surfaces.

A common classification is to divide features into two types: explicit and implicit. In an explicit feature the geometry is fully defined, whilst for an implicit feature, the feature is represented parametrically by attribute values and the full geometry must be calculated as required (Pratt and Wilson, 1985). Explicit features can then be classified into four main classes: through holes, protrusions, depressions and areas. Sub-classification as prismatic or rotational is possible according to the cross-sectional shapes. Tonshoff *et al* (1996) used explicit and implicit features in a modified manner, using a dual representation for each feature in order to integrate regular and free-form features into a unified classification.

Gindy [(1993), (Gindy *et al*, 1993)] has developed a taxonomy which is particularly suited to manufacturing representations. Features are treated as volumes enveloped by entry/exit and depth boundaries. A feature is defined by imaginary and real faces in its definition. The number of imaginary faces determines the External Access Directions (EADs) which can be used for process planning. Each EAD has a corresponding exit boundary status (through, no through) and type (open, closed). The number of EADs is used to group the classified features as either protrusions, depressions or surfaces.

Gandhi and Myklebust (1989) group features according to common topology, such that features sharing the same set of parameters are grouped together. An example would be the group of features which can be described by the parameters of a length and a radius, which includes “cylinder”, “disk” and “cylindrical plate”. An additional level of classification can be applied according to “form”, such as angularity, curvature, rotundity, straightness and circularity. This taxonomy is perhaps less logical in the need for a combination of two separate classification schemes.

Latif *et al* (1993) describe an object oriented feature taxonomy based on the definition of a base or stock feature which is then modified by the addition of addition of subordinate features. Features belong to either base, depression, protrusion or surface classes. The individual features are modelled parametrically as instances of the classes.

Vancza and Markus (1993) extend the modelling of features to include the concept of intermediate features, which exist temporarily during production but not in the final component. This approach is an attempt to handle certain difficulties in feature to process mapping, such as (i) multi-step processing, (ii) alternative production paths and (iii) processes covering multiple features. The authors contend that intermediate features are necessary to represent the alternative production paths available and to make process plans realistic: for example, features which are clustered according to quality requirements may result in all processing for that feature being clustered, whereas in a real plan only the finishing processes need be clustered.

These schemes must be measured against two requirements for feature taxonomies: A rigorous taxonomy is a prerequisite for the production of predictable analytical algorithms for engineering systems and secondly, the feature taxonomies and representations must support the generation of the geometry during design. Gindy's scheme is aimed at providing a structure which simplifies the generation of process plans and meets the first criterion very well for a particular analysis requirement. Latif's scheme similarly uses the vocabulary of process planning and is most suited to this domain. Butterfield's taxonomy is less specialised and suitable for an integrated product model used by many analysis systems, whilst the schemes of Pratt and Gandhi are more strongly aligned to the second criterion.

Case *et al* (1994) used Gindy's feature taxonomy to implement an integrated CAD/CAM system. A useful concept introduced in this research is that of compound features, which are defined as: "a group of sibling primitive features which it may be useful to treat as a single entity because, for example, they perform a single function or may be machined by the same manufacturing procedure".

2.4.2.2 Feature recognition

The process of feature recognition seeks to examine an existing geometric model and to recognise features according to a known library of feature types. The methods used to perform this task are highly dependent on the type of geometrical model: B-Rep models lend themselves to this approach since they are generally based on a graph model which can be parsed by the feature recogniser. CSG models are far more difficult, since a CSG model can often be decomposed into an arbitrarily large number of equally valid feature sets. The challenge in this case is to reflect the design intent in the feature set selected.

Lenau and Mu (1993) list five categories of feature recognition methods: (1) syntactic pattern recognition, (2) state transition diagrams. (3) decomposition approach, (4) CSG (set theoretic approach) and (5) graph-based approach. They make the point that the feature recognition approach is inherently unsatisfactory, since the CAD data does not originate in that form, rather from the designer. Hence any feature model that is generated in this way is inevitably a translation of a translation, with a resulting loss of accuracy of information content. One of the chief criticisms of the feature recognition approach is that it promotes a "wanton abandonment of design intent" (Case and Gao, 1993), such that what design intent is captured in the geometric model is not passed on to the features.

However, feature recognition researchers (Pratt, 1993) contend that these methodologies have many applications in product development. In particular, feature recognition techniques can be applied to the task of mapping features representation to another, to generate such features as manufacturing, assembly, fixturing, robotics, inspection, analysis and design features. In addition, the technique can provide valuable validation for feature based design systems.

2.4.2.3 Design by features

Design by features is the process of first building up the product model by adding features from a pre-defined library then building any geometric model from this feature information. Since the feature representation maintains all the attendant data relating to the feature model along with the geometrical information, there is no subsequent requirement for feature recognition. This has been proposed by many researchers as the most appropriate solution to the need to generate feature based product models.

The requirements of a feature based design system are set out by Case and Gao (1993) from the work of Pratt and Wilson (1985) and Shah *et al* (1988):

- The data supported must be sufficient for all applications that will use the database
- The mechanism for feature definitions must be flexible (generic) to allow designers to define features in any form, at any level for their own needs.

- The product definition system must provide an attractive environment for creating, manipulating, modifying and deleting feature entities. Feature relationships should also be defined.
- The design system should be able to integrate with different application software and the interface mechanism should be flexible so it requires minimum effort.

Case (1994b) describes a design by features implementation using Gindy's feature taxonomy to create feature based product models in a B-Rep solid modeller called Imaginer (Pafec, 1991). In this approach, the feature representation was maintained in a parallel data structure, although the author suggests that it would be preferable to redesign the solid modeller's structure to add the feature data if possible. An iconic user interface was used to select feature types to add to the model and to define relations between features. It is claimed that the iconic feature based interface proved a more efficient and robust means of specifying geometry than the underlying solid modeller. Further work from this project was reported by Wan *et al* (1995).

A design by features approach gives the system the capability of performing assessments on design decisions as they are made. An example of this has already been mentioned in the work of van der Net. One means of adding this functionality is through the use of knowledge based systems (KBS) attached to the product model. The work of Dixon (1988) is an example of this approach, where a KBS is used to monitor the design process to ensure that operations requested by the user are allowable and understandable to the system. Medland and Mullineux (1993) also employed a constraint based approach to feature-based design. They use manufacturing motion features (MMFs) to define part geometry by interacting with a stock material form. This work is designed to ensure manufacturability of parts. The authors also investigate the possibility of using MMFs to check the validity of feature recognition results.

The main criticism of design by features is that the limitation to a defined feature library will over-constrain the designer. However, this is equally a problem with feature recognition which will also fail if a design has features outside the existing taxonomy. Furthermore, it can be considered a benefit of the system that the designer is required to use standard solutions to problems by restricting the allowed geometry.

2.4.2.4 Feature Mapping

A technology which is related to both feature recognition and design by features research is that of feature mapping. In this process, a product modelled using one feature representation is converted into a alternative representation which is to be used for a particular activity.

Fu *et al* (1993) present a feature representation which is tailored to the requirements of feature mapping, or feature transformation. They specify the need to "support automatically the different ways specialists view the same object" as the main driver of the research. This approach adopts graph grammar representation of features, where a feature is a sub-graph of a design graph. In this work, a set of graph grammars are defined for each feature class which allows the feature to be transformed from one view to another by parsing it according to the grammar. A KBS approach is used to apply constraints to the graph to maintain representation integrity. Wong and Leung (1995) describe a feature conversion system using a dual feature concept. Features are either neutral features or application features. Neutral features are used as an intermediary stage to convert features from representations which are specific to particular applications. The representation is currently limited to prismatic parts. It is not yet clear whether the use of multiple feature representations will be more powerful than a single representation understood by all applications, but the use of application independent features appears to be the most promising means of transforming features.

2.4.3 Dimensioning and tolerancing of product models

Dimensioning and tolerancing are central requirements of product modelling systems. Most solid modellers, however, have difficulty in representing geometric uncertainty and particularly in displaying uncertain geometries (Kulkarni and Pande, 1996). In this section the different approaches to the modelling of dimension and tolerance information is explored, together with some of the applications of such model data. Zhang and Huq (1992) present a review of tolerancing techniques in all areas of product development. They identify six categories of research into tolerancing, which are: Dimensional tolerances chains, geometrical modelling in tolerances, statistical and probabilistic methods in tolerancing, tolerances based on analysis and synthesis, tolerances based on cost-tolerance algorithms and design methods. It is the geometrical modelling of tolerances concerning product modelling which is of interest for this work.

Requicha and Chan (1986) present a scheme for representing surface features and associated tolerances within a constructive solid geometry model. This approach treats tolerances as attributes, or properties of features within the model. The tolerance attribute is combined with the basic CSG feature to create a *tolerance zone*, within which the feature geometry must fit. This has become known as the solid offsets approach. An important weakness of this approach is that all tolerances are treated together and individual tolerances cease to be independent constraints (Turner, 1993). Farmer and Gladman (1986) have pointed out that this approach differs from the tolerancing standards.

Jayaraman and Srinivasan (1989) have proposed a *virtual boundary* approach, where the part is represented by a virtual boundary which represents the limits of the part size when the combined effects of all tolerances are taken into account. This approach suffers from the same disadvantages as the solid offsets approach.

Turner (1993), develops the feasibility space approach for tolerancing, whereby the solid model created by CAD represents the nominal part geometry. The actual manufactured instances of this model are represented as instances of a class of a variational model. The variational model is defined by specifying Cartesian points from the solid model, which are allowed to vary within constraints set by the tolerances. This approach has been applied to a system for automating tolerance assignment using a solid modelling system.

Kulkarni and Pande (1996) note that most solid model tolerancing systems use boundary representations and are therefore difficult to reconcile with feature based models and error prone. The authors propose a system (TMS) combining concepts of generated dimensions, direction vectors and dimension trees to facilitate a valid representation of tolerances in component solid models and overcome these problems. A generated dimension is a linear or angular dimension which is used to link two model entities in order to represent tolerances. Critically, it does not form part of the main solid model, representing only tolerance information. Direction vectors are defined as the common normal between parallel entities and reflect the direction of relative placement of two entities. Generated dimensions are grouped into dimension trees which have a common direction vector, forming a graph based model. Tolerance validity can be ensured by preventing loops from forming in trees through precedence relationships. The scheme can be extended from linear tolerancing to represent geometrical tolerances and to groups of features. Application of the model to a method for designing inspection equipment is given. TMS is a promising approach to tolerancing as it applies a unified methodology to all tolerance types. The drawbacks of the approach lie in the requirement for feature recognition from B-Rep models and with the separation of dimensional and tolerance information into separate graph models.

Dimensions and tolerances play a vital part in influencing the production cost, therefore it is important that they are set to the right level. Work has been undertaken to address the question of automating dimensioning and tolerancing. Zhang (1996) proposes a methodology for the simultaneous tolerancing of designs in order to integrate tolerancing into the concurrent engineering environment. This work seeks to define a general mathematical model for simultaneous tolerancing and to consider two cases in particular: worst-case and statistical. An attempt is made to link tolerancing to machining process selection using sets of "interim" tolerances. The proposed model allows the use of dynamic tolerance control, in which the measured tolerances achieved are fed back into the model to help the designer to redistribute tolerances if required.

Panchal *et al* (1992), developed an expert system to perform automated tolerance allocation (CATAP). The system performed feature recognition on a 2-D CAD drawing to extract features. Features on assembly drawings were interpreted to mating parts, the user was then prompted to specify the type of fit (e.g. interference) from which the appropriate tolerance values could be inferred according to the industrial standard. This system had many drawbacks, particularly in its failure to recognise redundant dimensioning, lack of integration and limitations to simple 2-D geometry. However, it provides a useful demonstration of the potential uses of feature models in tolerancing.

He (1991) proposed three objective functions for cost minimisation to be used in a computerised optimisation program which would determine the optimum set of process planning tolerances and dimensions to satisfy the specified design tolerances and dimensions and machining capability. The optimisation functions take into account machining cost, scrap and rework. This scheme operates on a single process plan, making it potentially useful primarily in the detailed design stage in conjunction with an automated process planning system. Several researchers have produced automated implementations of the tolerance chart method of assigning tolerances [(Ping, 1994), (Mittal *et al*, 1990)]. Typically these approaches use linear programming techniques to assign tolerances and dimensions to the intermediate stages of geometry in process planning. These tools have no application in design tolerancing, however.

Other researchers have developed methods to assess the manufacturability of specified dimension and tolerance levels (Mei and Zhang. 1992).

2.5 Process Modelling

The field of process modelling is concerned with capturing the production expertise of process engineers and with determining models for process factors which may not yet be fully understood by the engineers, with a view to supplying this knowledge to the whole product development team. The aim is to store artificial expertise which can be accessed by whichever engineer has a requirement for it and additionally can be built into automated analysis systems, whether they be DFx, CAPP or CE systems. Process models can be used both to analyse planned production and for comparison between measured performance and theoretical targets. They should contribute to the understanding, management and improvement of production.

Lenau and Alting (1992) identify a number of process modelling technologies: the morphological model, group technology, books of manufacturing processes, value control guide and constraint modelling. They point out that most sources of information on processes do not adopt a uniform method of description, making it difficult to compare processes and codify process knowledge. They propose a design oriented process model based on the following structure

- Basic transformation: Defining how it performs the transformation of input data to output

    - Reliability
    - Tool/Motions
        - Characteristic motions
        - Fixtures
        - Features
    - Materials
- Pre- and Post-processes
- Consequences
    - Cost
    - Environment
- Equipment/machines
- Product examples
- Company policy/strategies

The process model should incorporate information on the transformational capabilities, the resources requirements and the consequences. The way in which the process is used can be managed using the company policy model and product examples. Maropoulos (1993b) emphasised the need to evaluate the accuracy of process models, particularly when generic modelling techniques are utilised. In addition, specific requirements for improvements in detailed process models were identified:

1. Adequate materials technology
2. Feedback from process results
3. Compromise between optimal performance and data requirements to maintain sustainability in industrial environments.

Numerous researchers have developed process models for use in intelligent manufacturing systems. Esawi (1994) developed abstract cost models for a wide range of processes. These models were based on complexity ratings and weights of components. Flores (1993) compiled aggregate process models for machining based on cutting data supplied by tool manufacturers SECO (1997) and Sandvik (1997). These process models used the detailed calculations of cutting time, requiring all cutting parameters be known. The main thrust of this work, however, was to identify average cutting data which could be used for estimating purposes.

Allen and Alting (1986) published a classification of processes and accompanying models in a manual intended for student use. Processes were divided into shaping and non-shaping. Within shaping, three classifications were used: mass reducing, mass conserving and joining. Mass reducing processes were divided into mechanical (including machining), thermal and chemical categories.

There are a great many detailed process models which have been previously developed. Typically these models consist of an algorithm which interacts with a knowledge source. Most such models are constraint based. Detailed process models are usually only suited to use when a design has been fully defined and all the model's input parameters can be determined. There is a need for process models which can be used at aggregate level, when some design parameters are not determined and may be allowed to vary to improve the process.

2.6 Integration of manufacturing tools (CIM)

The requirement for an integrated set of support tools in product development has often been stated (e.g. design *toolkit*, engineering *workbench*, etc.). In order to achieve this aim, a number of enabling technologies have been developed. These include the drive towards standardisation of data models and transfer protocols, the development of distributed systems and the explosion of interest in the use of networks (Internet and Intranet) and associated tools.

2.6.1 STEP

The use of CAE tools within multi-disciplinary teams requires that the engineers operate on shared data models which represent all aspects of the design. Data transfer between different packages, which may perform similar or very different tasks, is required to support the existence of alternative competing CAE products in the market. The STEP (Standard for the exchange of product data) standard (ISO, 1994a) has defined a protocol for the exchange of product data across all engineering domains which is independent of hardware, software and process. STEP is based on the

EXPRESS data definition language (ISO, 1994b) and recommends the use of a neutral file format for the transfer of product data from discrete systems. An *et al* (1995) describe a CIM system architecture which implements the STEP standard. The STEP representation of objects (such as features, or points) models not only the geometry of the entity but also its behaviour.

2.6.2 Interactions between intelligent computer systems

Lu (1990) proposed a conceptual framework for concurrent engineering which relies on an integrated data model. This is based on the concept of multiple co-operating knowledge sources (MCKSs) which operate on a single data model or "blackboard". In this paradigm, multiple knowledge based systems which specialise in individual development activities operate concurrently on the product. To facilitate this approach, two problems must be solved: management of the co-operation between discrete knowledge sources and development of appropriate knowledge sources to cover all product development tasks. The latter problem is that faced by most CE systems, but the former has led to specific research. Other researchers (Brandon and Huang, 1993) have used to concept of *agents* to represent co-operating expert systems. A wider definition of agents may be used, covering knowledge based systems, other computer systems or even human experts. Research into intelligent agents has covered means of interfacing between agents and of negotiation between agents. Kannapan and Marshek (1992) have developed a methodology for the optimisation of design parameters shared between intelligent agents using negotiation graphs and the concept of utility. This approach depends on a detailed model of product functional relationships. Indeed, the MCKS concept as a whole is based on an assumption that products are fully understood and all aspects may be modelled. Whilst this will be true for re-design in mature technologies, much new product design involves the introduction of new concepts and technologies, which cannot be adequately described in knowledge based systems, therefore the MCKS approach would fail in these circumstances.

The concurrent engineering ideal of integration of customers and suppliers into the development team brings in a requirement for data transfer via Internet systems and the translation between different software and operating systems.

2.7 Systems for Concurrent Engineering

A number of research efforts have attempted to develop methodologies for the support of concurrent engineering with varying degrees of success. These projects have generally recognised a requirement for the development of an integrated computational architecture which provides automation and analysis tools for use by members of the product development teams. These systems provide important background to the development of the system described in this thesis. Whilst many systems have introduced useful concepts and provide valuable functionality, no single approach has been universally accepted. The system which has been developed in this project has drawn on elements from these systems. This section is divided into four sub-sections: The first three sections represent increasingly complex systems, from the more simple rule based approach to the complex integrated systems. The fourth section discusses systems which have been developed to investigate specific issues with the product development process (e.g. quality, design originality).

2.7.1 Rule centred expert systems

The expert system approach is characterised by an absence of a product model. The user interacts with the system to provide data for a set of rules which is used to assess the manufacturability of a design.

An example of a rule based concurrent engineering system is MPSS (Manufacturing process selection system) (Percyk and Meftah, 1997) which has a two stage approach to process and design selection. In the first stage, process selection is performed by calculating the *process index* of each process, according to the degree to which the design meets the criteria set out in a number of design for manufacture rules. Six degrees of fit are recognised for each rule, so that the process index generated is expressed as a fuzzy number. The importance of each rule is varied for each material and process according to coefficients. The second stage is to select a single process for each design alternative and to calculate *design indices* with which to compare these alternatives. The system has been developed for casting processes, initially, rules being checked interactively by asking the designer to "mark" the design.

The DAISIE system (Ishii, 1993) applies case based, experiential reasoning to determine a design compatibility index which shows the degree to which the major elements of a design match the specification and life-cycle criteria. The design is analysed in an interactive procedure, with the user answering detailed rules about the design (e.g. pick most similar shape of a forging, or for a rib feature, enter thickness, depth etc.) from a design compatibility knowledge base (CKB). The CKB models the set of relations between the specification space, design solution space, life cycle process space to produce a normalised measure of compatibility. A drawback to this approach is that the system has no in-built product model and will only assess what it is told of the design. It is possible that features which are not assessed may jeopardise the whole design.

2.7.2 *Model based expert systems*

Model based expert systems use an abstract model of the physical world to reduce the number of specific rules which are required. A design model is generated which is then assessed using a set of concurrent engineering rules.

Colton (1993) describes an intelligent design system (IDS) which used a model-based expert system approach to represent mechanical products. The system used a 2-D model of the 3-D world and simplified parts to just three materials, three processes and five features (e.g. holes, slots an ribs). To provide additional functionality, standard parts such as gears and motors were available in multiple sizes from a database. This system used physical models for torque and force to calculate strength requirements for fasteners and identify manufacturing violations.

2.7.3 *Integrated systems*

The integrated system seeks to provide a concurrent engineering system as a unifying module in a CAE environment. Typically an integrated CE system links CAD functionality with CAPP to perform manufacturing assessment. The goal of these systems is to integrate all life-cycle issues into a single environment.

Abdalla *et al* [(Abdalla and Knight 1994), (Abdalla *et al*, 1995)] have developed a KBS for automatically assessing component designs for manufacture. In this system, a rule based feature recognition system interfaces with a solid model (Pro/Engineer) to develop a feature based representation. The features on this model are then assessed individually using a process knowledge base. The system can identify feasible processes and estimate process capabilities based on the feature tolerances. Relative cost values for each process are also produced. This system suffers from a number of drawbacks, in particular, since features are considered one at a time, a high number of processes will be suggested. Also, the system assumes a single process is used for each stage and does not calculate actual costs to assess which process to use.

Meerkamm (1993) describes the Design system MFK, a prototype "engineering workbench" which combines functional and geometric design (synthesis) with a multi-functional analysis system. The synthesis module of the system has four elements: Geometry, Technology, Function and Organisation. This is more than just a CAD modeller, since it allows the user to specify the product structure in conceptual terms and to model the functions of components in terms of e.g. forces. The analysis module performs a checking function on the design and incorporates a knowledge base for production with links to external analysis engines such as FEA tools for specific checks. When problems with the design are identified, the system can suggest changes and make modifications if required. This system comes close to the required goal of an integrated product development tool, although it does not have universal coverage of product development activities. The system stresses the design aspect of development and therefore ties the design towards standard process paths.

Su and Wakelam (1997) describe an integrated CE system which uses a combination of AI techniques to cover many phases of product development. The system provides a link between a CAD system (Pro/Engineer) and specialist CAE tools using an expert system. Design selection is performed using an artificial neural network which matches specification information to standard alternative design concepts. Genetic algorithms are used to automate the training of the neural networks. This approach is suitable for building systems which focus on one particular product group (such as mechanical transmissions in this case) since the alternative design concepts need to be standardised. It is less useful for the support of creative design when original concepts are introduced, also for generic implementation since it requires detailed information about the product domain.

2.7.4 Specialised systems

Mezgar and Kovacs (1993) describe a CE system which integrates product design, process planning and facility design with a goal of improving quality assurance. An expert system approach is used, using the blackboard architecture, where multiple discrete expert systems interact with the same product model to assess the model in parallel. The system uses a real-time expert system to link a feature-based CAD model with CAPP. The resulting process plans are passed to a simulation system combined with a layout planner (SIMAN/Cinema) to design the factory layout at the same time as the product design. The stated future goal of the research is to integrate real-time shop-floor control with the product development process. By linking factory and product design, the system gives the developer the opportunity to consider whether it is better to modify the design to suit the factory, or to modify the factory to suit the design: this is close to a true CE application.

A number of process selection systems have been developed which can be useful in CE. Kristensen and Lenau (1992, 1993) describe MADED, a process selection system which is aimed at designers in order to explore alternative production methods and to trigger ideas for new design configurations. The system employs a twin strategy search of a process database, one search being based on the process characteristics, the other on the component. The component search looks at the way in which similar components were manufactured, using a search based on parameters of name, function, shape (uses skeletons, [Lenau and Mu, 1993], configuration/complexity, functional surfaces and features), material and process (e.g. quantities etc.). The component based search has been developed as means of overcoming the difficulties of linking process models directly to component geometry models. The process search is a direct search based on requirements such as tolerances, sizes and volumes. Processes can be evaluated for cost, life-cycle consequences, etc. MADED has been tested for primary processes such as casting. Esawi (1994) describes a method of systematic process selection for mechanical design, to be used in early design. This method is based on a cost model for processes, which be used to generate a graph of process against cost for all feasible manufacturing processes. The cost model evaluates data such as material, production volume and quality levels. The process selection element of MPSS, (Percyk and Meftah, 1997) produces similar output using a different procedure. Haudrum and Alting (1993) have also considered methods for the integration of process selection during design. Process selection is only a single part of concurrent engineering, however these systems can be useful to provide starting points for CAPP systems.

Gindy *et al* (1994) have developed a system for equipment selection based on capability models. Alternative process plans for components are expressed as TSFs (technological solutions at feature level) which consist of sets of form generating schemas (FGS). A form generating schema combines a cutting tool, a motion set and nominal technological output (e.g. tolerance range). A selection procedure based on a clustering algorithm is proposed to determine the optimal machining cell to use for manufacture based on the capabilities of the machine tools.

2.8 Conclusions

This survey leads to a vision of a new manufacturing technology support strategy for concurrent engineering environments using integrated yet distributed IT tools which provide analysis and assessment functionality to many disciplines within product development and allow a broader range of concepts to be investigated in more depth, through the use of artificial expertise, than is possible in traditional manufacturing product development.

Thus the CE goal of front loading the design process can be achieved, problems identified early and solutions found and changes made before detailed work is carried out and high costs are committed.

The main area where there is a lack of progress in reaching this vision is in the provision of adequate production analysis for the early stages of design. This is therefore the principal target of the research.

Chapter Three

System Overview

This chapter presents an overview of the prototype computer system which has been developed to implement the proposed concurrent engineering methodology for product development.

3.1 Introduction

It has been established that there is a requirement for enhanced technology support for concurrent product development. It is the thesis of this work that an integrated computer system operating at an aggregate level is the best means of providing support in the product development process in a concurrent engineering environment. An aggregate product development methodology has been developed and tested through the use of a prototype computer system which implements the proposed methodology. This prototype system is an integrated decision support environment which performs automated product design and factory assessment functions on early products. The system is called the Concurrent Engineering Support System (CESS).

The methodology is defined in this thesis through the specification and functionality of the prototype system, CESS. It is important to note that the development of CESS itself is not the purpose of this work, rather the system was developed in order to test the theories which have been applied in its development. This chapter will discuss CESS from a systems viewpoint, identifying the specifications of the system in terms of the tasks which are required. The structure of the system will then be outlined, describing the main system elements, each of which are detailed in subsequent chapters. Finally, both the tools which have been used to develop the system and the important development issues of object oriented modelling and expert systems are explored.

3.2 Specification of the Concurrent Engineering Support System

The CESS is a computer aided engineering tool which is targeted at filling the perceived gap in support for product development at the early stages. The primary requirements for designers during this stage of product development is the analysis of the ability of a given design to perform its required function. Design is always led by the requirements of the product function. For concurrent engineering, however, it is important to ensure that the designers are also able to consider the manufacture and subsequent life-cycle issues of the product. The manufacturing constraints should be considered by the designer along with the product performance constraints.

Whilst the design of a product is the principle influence on the manufacturing process, the designer is not typically an expert on production methods. Therefore, in order to properly consider the manufacturing consequences of a design, the following questions should be addressed: (i) Can this design be manufactured? (ii) What is the manufacturing cost for this design? Whilst it is usually possible to manufacture any given design, the cost of doing so may be unacceptably high. These questions lead to the definition of the *manufacturability* of a design, which is an indication of the suitability for production. Manufacturability can be measured in many ways (Chu and Holm, 1994), the most valid of which is in terms of cost that includes the effects of other indicators (for example the quality can be expressed in terms of cost [Taguchi *et al*, 1989]). Factors contributing to product cost include: material, labour, machine tool depreciation, tooling, energy, cost due to ensuring quality, storage (space provision and investment cost) and transportation cost. Alternative measures of manufacturability which will be useful include the manufacturing lead time for the product and the quality level which can be achieved for the product. When combined across the entire product range, such manufacturability indices can give an indication of the manufacturing agility of a company.

In order to assess the manufacturability of a design, it is necessary to identify possible means of manufacture, then to check the implications of the use of each of these alternative methods thereby arriving at costs for each alternative. CESS achieves this through the generation of Aggregate Process Plans. In order to operate during the early design stages, a concurrent engineering support system must fulfil a number of criteria:

1. Ability to represent alternative design concepts during conceptual, embodiment and detailed design stages.
2. Ability to perform automated aggregate process planning.
3. Ability to assess alternative aggregate process plans in terms of manufacturability criteria.

These three criteria lead to more detailed requirements about the structure and elements required in the system. In particular, in order to perform aggregate process planning and assessment of the production routes generated, it is necessary to provide the following knowledge within the system:

- Production process expertise.

  To perform automated process planning the computer system must capture process knowledge, including the geometries which the process can produce, the materials for which the process can be used, and rules for selection of both process parameters and calculation of input and output criteria.

- Manufacturing resources available to the company.

  Aggregate process planning should be integrated with the manufacturing capabilities of an individual company. This implies that the system must have access to appropriate data on the factory resources, including the machine tools available, the layout of the factory and cost rates for machine and labour time.

3.3 Functional Description

The functions of CESS can be broken down into a number of separate modules which are linked together to provide the overall manufacturability assessment function. In addition to the design assessment the system must provide various support functions. In particular, it is important that the system allows the designer to browse and modify the current design model in order to compare alternative design configurations. This section will discuss the structure of CESS in terms of the functional tasks which it performs.

3.3.1 Design specification

The user of CESS is able to enter a description of the product design and create a model of the product within the system. This model can be modified through editing existing information, deleting information and adding new structures. CESS allows the designer to build new designs based on existing standard parts through loading previously saved component and assembly data files into the current design. It is recognised that a fully functional concurrent engineering support system would provide an integrated link to a three dimensional CAD system. The functionality envisaged would consist of an interpretation module which would extract data from the more detailed solid model into an aggregate product model. For an initial conceptual design, however, the use of a solid model system is not always appropriate; the CESS product model allows the representation of design information on components which cannot yet be drawn because they are not fully defined. Along with a means of editing designs within the system, this module also provides the necessary functionality to load and save current design ideas to files on disk.

3.3.2 Factory resource browser

This function allows the user to investigate the details of the factory database. This may be used in order to tailor the system's analysis through the selection of a particular cell, cluster of cells or cluster of machines which should be considered for the manufacture of a component. The factory resource browser allows the user to edit the resource model to ensure that it is up to date.

3.3.3 Aggregate Process Planning

This is the principal function of CESS: the manufacturability of a given design being assessed through the generation of aggregate process plans, which include estimated manufacturability criteria. The aggregate process planning function is divided into a number of stages. The main requirements for aggregate process plans are the selection of process, selection of machine tools to perform the process and the sequencing of the processing steps. The aggregate process planning function of CESS is a generative automated process planning system operating at an aggregate level. Plans are detailed enough to include multiple stages of processes (i.e. roughing and finishing) and the selection of individual machine tools, however detailed process parameters and tooling selection is not performed. Aggregate process data stored in object oriented databases is combined with individual process equations, feature and machine parameters to calculate the manufacturability indicators.

3.4 System Architecture

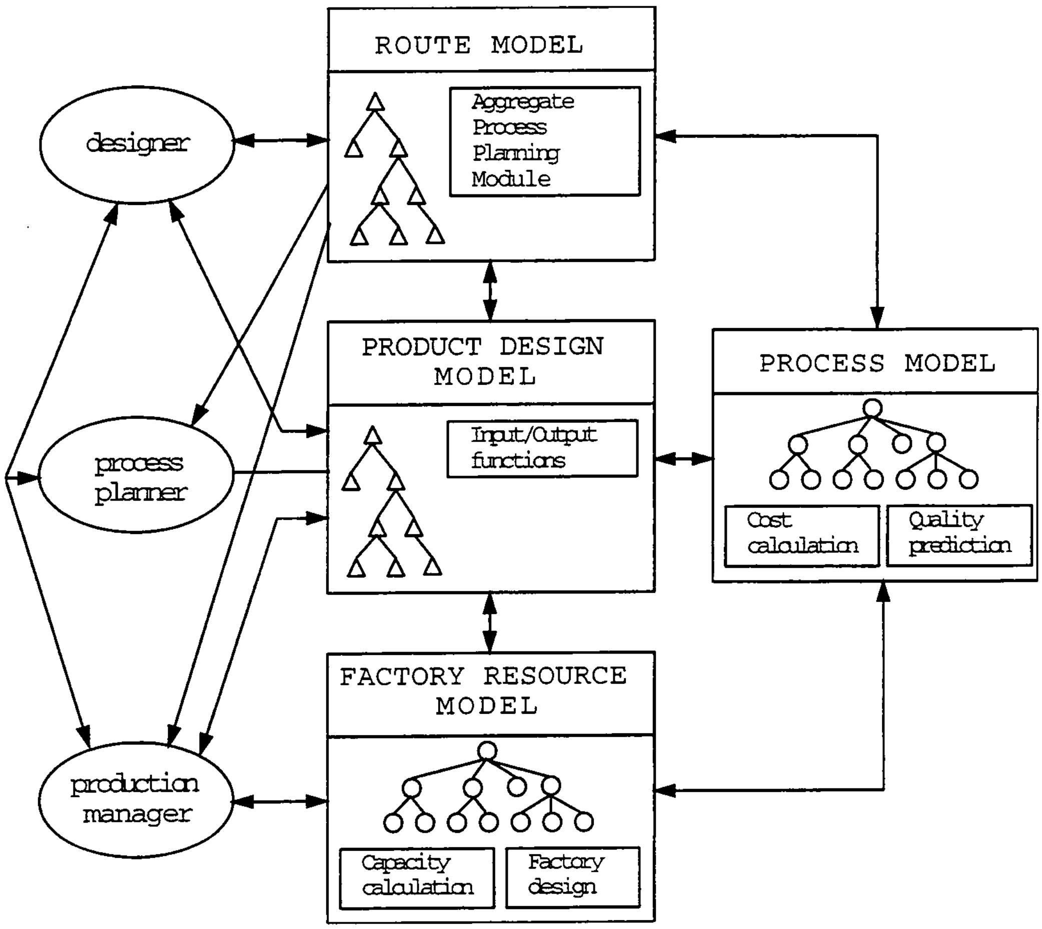


Figure 3.1: System architecture of CESS

The concurrent engineering support system is organised as an event driven system which allows the user to construct and modify a number of structured models. Specifically, these models represent the current design idea being studied, the factory resources available and any aggregate process plans which have been generated. In addition, the system has an object oriented model of production processes which can only be modified by the administrator. In normal use, the product developer would use the design editor function to enter the information available about the proposed product design, make modifications to the factory resource model with the factory editor function, and then analyse the manufacturability by running the aggregate process planning function. The architecture of the system is shown in Figure 3.1.

Since CESS is designed as a decision support system, the functions do not feed data back directly into the databases which are used to generate the individual models, instead the outputs are provided to the human users, who have control over the way in which the system databases are updated. This ensures the integrity of the system and promotes the engineer's understanding.

3.4.1 Product Model

The product model is an object oriented representation of the product structure. Products can be represented at component and assembly levels. The components are modelled using a feature based solid modelling approach which is compatible with the latest CAD systems and highly suited to manufacturing planning. The product model can represent a high degree of detail when necessary, including dimensional and tolerance information, whilst both retaining the ability to simplify the data in the early stages and allowing the representation of incomplete geometry descriptions. The product model is detailed in Chapter 4.

3.4.2 Process Model

The process model is a generic knowledge base of production process engineering using an object oriented structure. It is based on a hierarchical classification of process types (after Allen and Alting, 1985). A process archetype (superclass) defines the required functionality which each individual process class has to supply within the system. Required attributes of the detailed process classes include a list of feature types which the process can be used to manufacture and a list of the possible machine types which can be used to carry out the process. Process class functionality includes methods to enable the system to calculate the time and cost required to manufacture features using the process. When objects are instantiated within the system as members of a specific process class they inherit the functionality of the class, enabling the system to define the behaviour of the objects. These objects form part of the route model, discussed below. Further details of the process model are discussed in Chapter 5.

3.4.3 Resource Model

The resource model is an object oriented model of the factory resources available to the manufacturing planner. The factory resources are resolved to the machine tool detail level. Three classes of object are recognised in the factory resource model, namely factories, cells and machine tools. Factories and cells are defined as logical groupings of cells and machine tools, respectively. The ability to model more than one factory within the system allows for the potential use of the system in benchmarking studies or in make or buy decisions.

The machine tool objects are instances of a machine tool classification which uses a network structure to enable the modelling of the great variation in machine tool configurations which are available. Machine tool attributes include operating power, efficiency, component size envelope and speed constraints. The current implementation of the machine tools model covers turning, milling and drilling machines. The resource model is discussed in detail in Chapter 6.

3.4.4 Route Model

The route model allows the system to store and manipulate aggregate process plans. This model is generated by the system through the automated process planning function. The route model is closely linked to each of the other three data models, representing as it does the integration of the designed components with the processes used to make them and the machine tools which perform these processes. The route model is discussed in Chapter 7, which also describes the aggregate process planning (route generation) functionality.

3.5 Development methods

CESS has been developed using Nexpert Object, an object oriented knowledge based system environment designed for the rapid prototyping of artificial intelligence based computer systems. This system was chosen because it provides the required object oriented modelling ability along with a powerful implementation of the knowledge based system. In addition the environment provides an integrated library of routines for the development of graphical user interfaces. This allows the development of a windows based interface using either FORTRAN, C or C++. In this project the graphical user interface has been implemented using the C language.

3.5.1 Object Oriented Analysis

Object oriented analysis (OOA) is a technique for modelling and understanding of complex systems. Coad and Yourdon (1991) describe object oriented analysis as "the challenge of understanding the problem domain, and then the system's responsibilities in that light". The key to understanding complex systems is to decompose the system into the manageable pieces which can be more easily understood. Traditionally, systems have been decomposed on the basis of algorithmic decomposition, which breaks the processes down into individual steps. In object oriented decomposition, the system is decomposed according to the key abstractions in the problem domain (Booch, 1991). Thus, instead of a set of process steps, the system is represented as a set of objects which are described in terms of their properties and behaviour. The advantages of OOA lie in the benefits of abstraction, encapsulation, inheritance and organisation methods. Abstraction allows the analyst to ignore those aspects of the system which are irrelevant and concentrate on the important factors. Encapsulation relates to the practice of hiding the complexity of an object from view when looking at the wider picture, thus reducing the complexity which must be handled at any one time. Inheritance allows the analyst to express commonality amongst objects, by defining attributes and behaviour (services) to classes to which several objects belong. The objects inherit the attributes and services of the parent classes, thus sparing the definition of each separately. Coad and Yourdon identify three pervading methods of organisation which are inherent to OOA: "objects and attributes", "wholes and parts" and "classes and members". Each of these enhances the understanding of the system and leads to a more complete description.

Within this thesis two different schemas have been used to represent object oriented models. Whilst in general a single representation scheme might be thought to be more consistent, there are advantages to using a mixture of two styles. The first scheme is that adopted by Coad and Yourdon (Figure 3.2). This representation highlights the encapsulation of data within objects and emphasises the relationships of wholes and parts. It is particularly good for representing the details of a class structure defining objects which are sub-objects of others, as shown in the figure. This system has weaknesses, however. In particular it is difficult to represent multiple objects belonging to the same class and to represent objects which are instances of more than one class (the concept of *multiple inheritance*). In these cases, the object/class model cannot be represented without showing the same class or object more than once on the diagram, which is confusing. In these cases, therefore, a second representation scheme has been used which is more flexible.

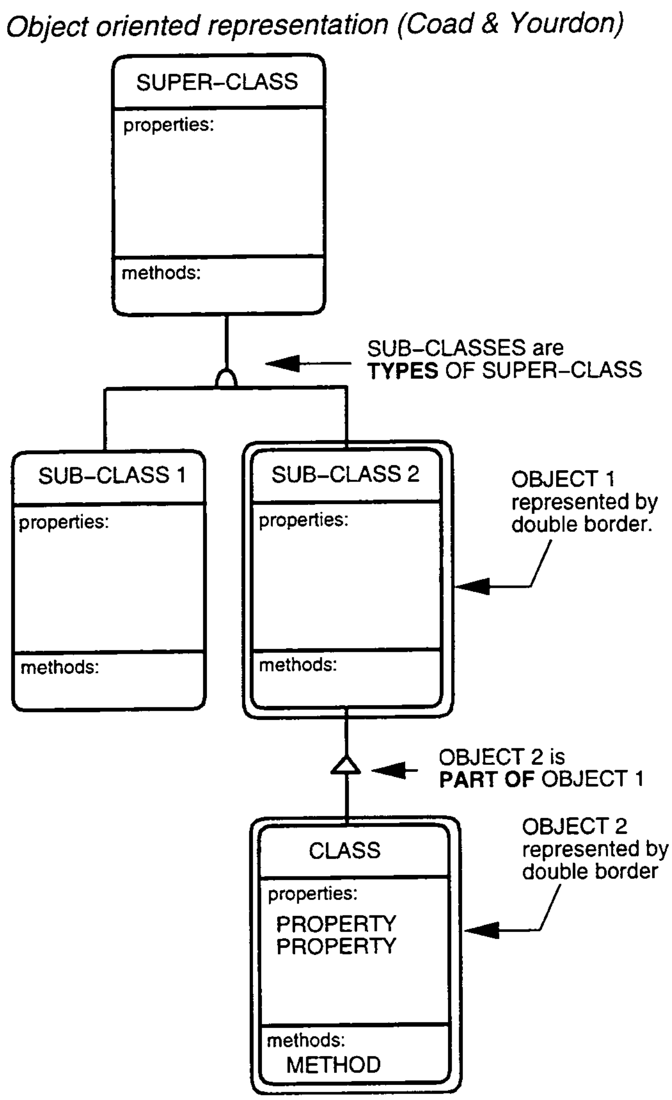


Figure 3.2: Coad and Yourdon schema for object model representation

The second representation scheme is a more basic method adopted from the manuals of the software development system used, Nexpert Object (Neuron Data, 1995). In this representation, different symbols are used to represent classes, objects, properties and methods (Figure 3.3). This representation uses a separate symbol to represent the objects and classes. It is thus easy to represent two objects which belong to the same class, or a single object which is an instance of two distinct classes. The main drawbacks of this approach are that it results in larger diagrams, making it difficult to represent complex situations, and that message paths between objects are not represented.

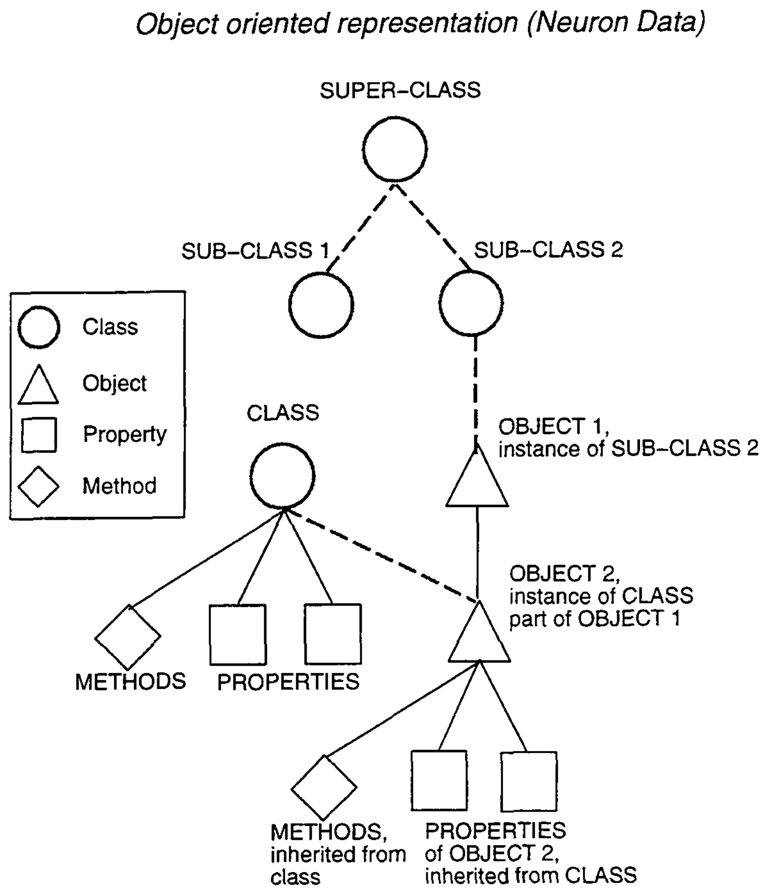


Figure 3.3: Neuron Data schema for object model representation

3.5.2 Object Oriented Programming

OOP is the development of computer systems based upon models generated through OOA. This is a particularly powerful programming approach which has become the most common type of computer language in use today. Examples of OOP languages include LISP, C++, Object Pascal and Java. Object oriented programming is particularly suited to manufacturing applications since the data models relate closely to real world objects. Furthermore, object orientation supports the maintenance of models at multiple levels of detail. This is particularly useful in the modelling of a product throughout its development, since the initial model will be far less detailed than the final one.

In an object oriented program, the data is stored as objects which are members of one or more type classes. The type classes to which the object belongs determine the functionality which applies to that object, since the functionality of the program is stored as "methods" attached to the classes. These methods are sets of instructions which are executed by the sending of "messages" to the object or class to which the method belongs. An object oriented program operates by the sending of messages from one object to another, causing methods to be executed which may in turn generate further messages. An important concept of OOP is "inheritance", which is the mechanism by which the functionality of the program that is stored in the classes is propagated to the objects created during the program execution. Objects (and classes) may inherit methods and properties from their parents. Thus, all the functionality which is required to be associated with an object may be assigned through the membership of particular classes. Objects can be members of several classes and have multiple parent objects. Thus, an object oriented model stores information not simply in the properties of the objects but in the linkages between the objects and the relationships which are created. The use of multiple classes for single objects gives the programmer a finer degree of control over the system behaviour. This programming method is suited to the generation of product models in particular, since the class of the objects within the model can be changed during the development process so that more detailed methods can be applied to the increasingly detailed product design.

3.5.3 Knowledge based systems

As stated previously, the Nexpert Object language used for the development of CESS is a knowledge based system engine. A knowledge based system is a computer program which systematically encodes human expertise in a particular field into a data retrieval mechanism allowing automated interrogation of the data to solve given problems. Use of a knowledge based system structure is an approach well suited to the design of a decision support system because the process of decision making can be made transparent to the user so that the reasoning behind each system suggestion can be traced. This enhances the reliability of the computer system since any errors which are made can be picked up.

3.5.4 Hybrid systems

A hybrid system is one which combines the elements of two or more alternative programming systems. Nexpert is an example of a hybrid system. The chief advantage of this hybrid system is that it allows the flexibility of modelling and ability to generate generic data structures characteristic of OOP, with knowledge based system functionality such as inferencing, which is highly suited to the encapsulation of engineering knowledge such as process planning expertise.

3.6 User Interface

CESS is implemented on a UNIX platform using the X-Windows environment to provide a graphical user interface. The user interface is based around a main development manager window which allows access to each of the functions of the system. The functions of the system call up additional windows to provide specific information such as the product model browser and the factory layout browser. These windows are programmed to be modeless, i.e. the program focus can shift to any of several open windows, allowing the system to be used in a non-linear fashion. The window controls are implemented in C, with functions which read the data from the knowledge base and use it to populate the elements of the windows. The user interacts with the window data and this is then passed back to the knowledge base which processes the data (Figure 3.4).

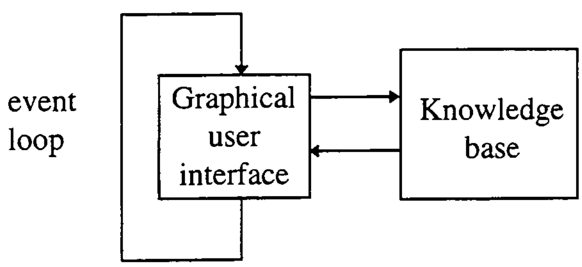


Figure 3.4: User interface architecture

The structure of the user interface is shown in Figure 3.5. The interface is based around a *main window*, which provides an overview of the current product, and can be used at management level to specify the resources which are to be designated for each product to be made. From this window, the remaining system functions can be accessed. The *business strategy window* is used to set the system variables which are used by the system algorithms. These include both the priorities between reducing costs and improving quality and the setting of wage rates and other cost factors. The *system manager window* gives the user control over system variables and allows management of the databases.

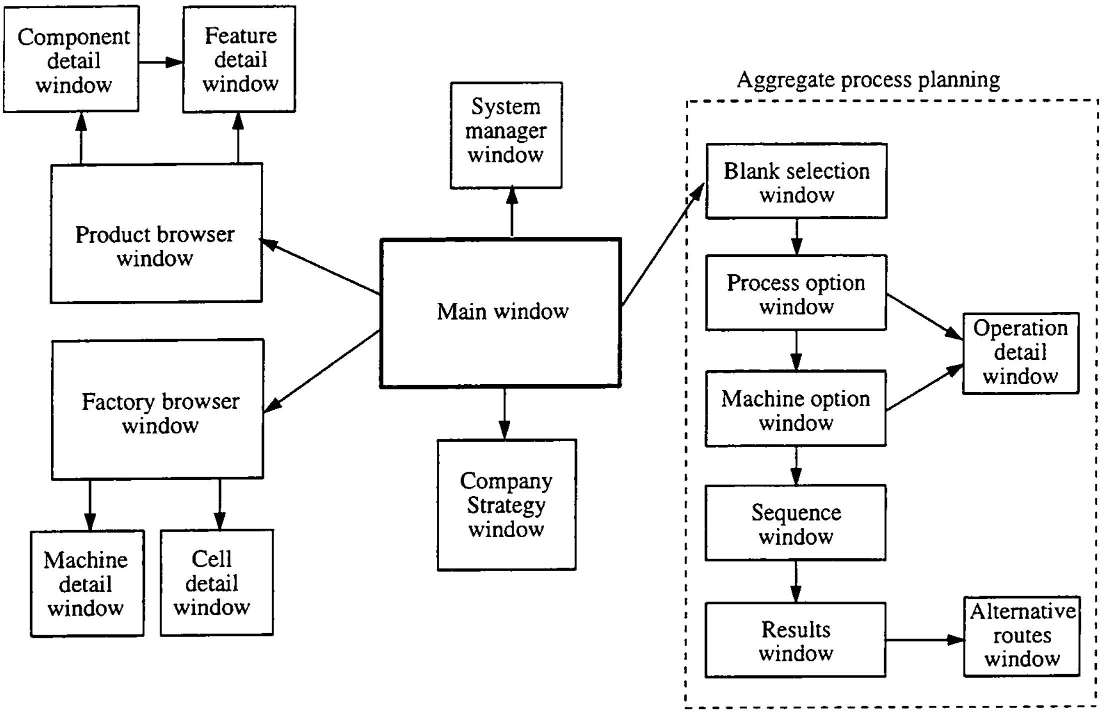


Figure 3.5: User interface windows

The *product browser window* is the primary means of design specification at present. This allows the user to alter the design model in any way, including the loading and saving of product definitions and the modification and addition of features. The product browser window consists of an acyclic node diagram of the product, in a main window, with an overview window to allow rapid navigation of this area. Modifications can be made through the use of context sensitive menus attached to each of the nodes on the product browser, which call up the appropriate input and output windows. The *factory browser* allows the user to analyse the state of the resource model. It displays the factory layout with the machines depicted in colour coding according to the cell membership. Specific machine information can be accessed by pop-up menus from the machine icons. The aggregate process planning functions use a succession of windows for each stage of the algorithm. Once the system has generated a set of aggregate process plans, the *alternative plans window* can be used to display each alternative plan on a graph of time against cost so that the plan displayed in the results window can be selected.

Whilst user interface design is only peripheral to the research objectives of this work, it is unfortunately a necessary requirement if the system developed is to be tested properly. A large amount of work has been done on developing the interface, however this work will not benefit this project alone, since it is to be used in the testing of other research tools currently under development.

3.7 Conclusions

This chapter has discussed the overall structure of the work and the computer system developed (CESS). The work may be divided into four models which store the data and the functionality of the system. The next four chapters detail each of these models in turn after which an example of the system as it would be used is given in the results chapter.

Chapter Four

Product Model

4.1 Introduction

This chapter describes the Aggregate Product Model which has been developed during this project in order to satisfy the requirements of the proposed process planning and design methodology. CESS implements the aggregate process planning function of the AMD architecture. This defines the requirements of the product model, which are set out in section 4.2. The next section discusses the development of a product model which satisfies these requirements and presents an overview of the elements and structure of that model. In section 4.4, the details of the feature classification which is used are defined, whilst in section 4.5, the dimensioning and tolerancing schema is detailed. This is followed by an example which shows the product model in use with a real mechanical assembly. The chapter concludes with a discussion of the implications of the use of the aggregate product model and some conclusions.

4.1.1 Definition of a product model

A product model is a representation of the intended physical product. This model may represent any specified amount of detail. In particular, a product model may store geometrical and functional information about the product. For an aggregate product model, information stored is restricted to the geometrical domain and material type.

4.2 Specifications for an aggregate product model

In order to develop the product model for CESS, it was necessary to identify the requirements of product models for aggregate process plans. These requirements come from the functions which must be provided by a product model and the varying situations during which the aggregate product model is to be used, i.e. aggregate product models are required to cover product designs during all stages of development. Product data differs in quality of information as well as in quantity of detail from conceptual to detailed design. In conceptual design, decisions are made between alternative function structures which could meet the specification of the product. This determines a basic list of components and their principle attributes. It is neither possible nor desirable at this stage to produce a geometrical representation of the part, since this will depend on factors yet to be considered. At this stage, however, the developer should be able to make some assessment of the relative manufacturability of alternative conceptual design options, in order to select the most appropriate.

In embodiment design, the components or the product are designed in more detail by mapping the functional requirements of the product onto particular features of the components. The key functional dimensions of the components are identified as parameters in the product performance model and the desired values are determined. At this stage it becomes possible to produce a schematic representation of the component geometry. However, much of the geometry of the parts will be indeterminate, since it is not related to the function of the product and can be left open to be assigned according to optimisation of other product factors, e.g. manufacture and reliability. At this stage the designer should be able to call upon DFx assistance to determine suitable values for the indeterminate geometry. Further, there is a requirement for the assessment of the manufacturability of the components and identification of production processes which will determine the detailed design characteristics. For example the optimal design will be different if it is to be made from a casting or machined from a solid billet. In the detailed stage of design, the full part geometry is specified, along with the variability which is permitted to that geometry. At this stage the requirement is for detailed drafting of designs, visualisation to allow verification of form details, access checks and data for detailed process planning and the generation of NC machine code. For the majority of these tasks, a full solid model of the product is required.

With this understanding of the product modelling through the development cycle, it is possible to list the specifications of an aggregate product model, which is suited to use in the early design stages. The requirements of an aggregate product model (APM) may be categorised into two main purposes:

1. The APM representation must support the transition of design data from the uncertain, conceptual stage through to the detailed design stage. To achieve this, the model must:

- Support the representation of incomplete data during the conceptual design stage.
- Support uncertainty in dimensioning component features in the embodiment design stage.
- Support the addition of new information and modification of the model at any stage of the design process.
- Support detailed product design information where appropriate, particularly in the representation of quality requirements (tolerances) which may be available when used for re-design work.

2. The APM must represent the design in a format which is suitable for integration with Aggregate Process Planning and with early design analysis systems. This means that it must:

- Store only the critical information about the product to reduce processing requirements.
- Use a structured representation built on a generic component model.
- Maintain the structure of data objects representing the features throughout the design stages.

- Use homogeneous representations for product quality based on industry standards for dimensioning and tolerancing.
- Represent assembly connectivity at feature level.

Whilst a number of alternative product models have been suggested by researchers (see section 2.4) for use during product development, the author feels that none is yet available which has sufficient flexibility and ease of manipulation to meet the above requirements for concurrent product development.

These requirements lead to the selection of an object oriented product model which uses feature based solid modelling techniques to define components. There are three distinct aspects which must be addressed in the development of an aggregate product model: representation of assemblies, component definition and dimensioning and tolerancing.

4.2.1 Assembly representation

At the basic level, the product can be considered as a set of interacting components. Simple products may consist of just a single component, whilst complex products consist of many levels of sub-assemblies and can include hundreds of components. An important feature of the APM is the ability to represent the logical grouping of product components into assemblies and sub-assemblies. At this level, the product model resembles the product bill of materials. When representing assemblies, the way in which components are connected together must be stored. The CESS product model recognises that assemblies may be created by reversible fastening processes, or by permanent joining processes such as welding or the use of adhesives. Connections between components are represented using assembly features relations.

4.2.2 Components definition

When seeking to represent the product design in conceptual and embodiment stages, it is important to recognise that the design will have many undetermined aspects. Design theory suggests that the best designs are achieved when each decision is left as late in the process as possible (design by least commitment) since this imposes the minimum number of constraints on every other decision. The challenge for a manufacturing assessment system in this environment is to provide a design model which can represent the design including the undetermined variables, and can perform analysis on this representation however much data is available. This leads to the identification of two requirements of the model:

- A flexible product model which allows the design to be changed easily
- A model which can represent conceptual designs and detailed designs with the same object constructs.

This leads to a model based on a simple geometry which can be refined by the subsequent addition of details. The most suitable modelling system for this approach is a feature based representation. When a component is described in terms of features, the principle is that there exists a set of distinct geometrical constructs of which the component is the sum. Features are thus not invented for a particular model, but exist naturally on the component and are there to be recognised.

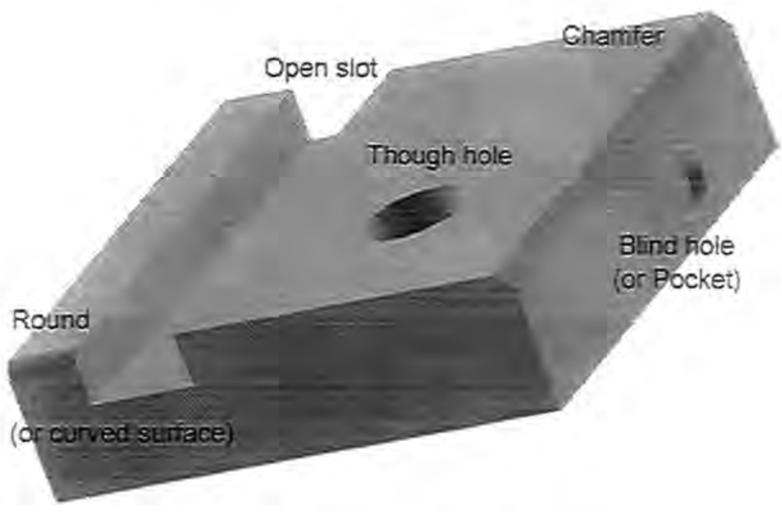


Figure 4.1: Feature classification model

Features can be categorised into three main types: functional features, upon which the operation and performance of the product depends (e.g. piston bore); manufacturing features, which are introduced to enhance manufacture (e.g. fillets on castings) and aesthetic features, which improve the look and feel of the product (e.g. chamfers on edges). In the conceptual stage, features usually represent only the requirements of the component function, whilst in the detailed stages features desirable for aesthetics or manufacturability are added (Table 4.1).

Table 4.1: Staged introduction of features during design

Design stage	Features which are added to model
Concept	positive feature, major functional features
Embodiment	minor functional features, manufacturing features
Detail	minor manufacturing features, aesthetic features

These stages of product development are shown for an example component, an internal combustion engine valve, in the following figures. In Figure 4.2, the conceptual stage of development is shown. A sketch of the design at this stage is accompanied by a schematic of the product model objects: *component*, *features* and *feature relations*. The object structure will be further detailed later in this chapter. At the conceptual stage, the designer is interested in the principle of operation of the valve. Thus, it can be represented by its key functional features. These are, the shaft cylinder, the cover cylinder and the sealing surface which is a chamfered edge on the cover cylinder. The aggregate product model consists of these three features. The presence of these features indicates the key parameters of the component.

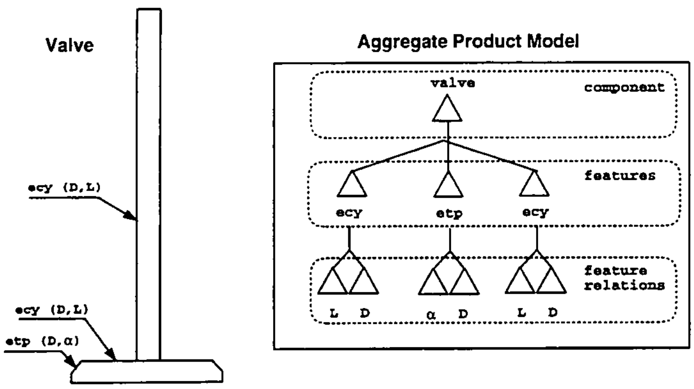


Figure 4.2: Conceptual model of the valve component

At the embodiment stage, the remainder of the functional features are considered, along with the major manufacturing features (Figure 4.3). Some features may be considered to fall into both categories. For example, the profile on the lower end of the valve is required both for manufacturing purposes (enhances forging process), and for functional reasons (increased structural strength and better fluid flow).

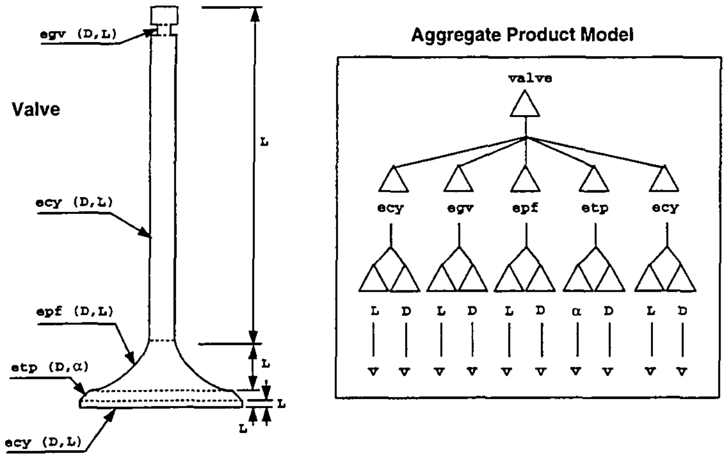


Figure 4.3: Embodiment model of the valve component

At the detailed stage, the component is fully specified with dimensions and tolerances for all components. The detailed product model shown (Figure 4.4), contains too many feature objects to show easily in the space allowed. However, the key elements of the model can be seen from the figure. In particular, the concentricity tolerance between the sealing surface of the valve and the shaft is displayed. In addition, the tolerance boundaries on each parameter value have been specified.

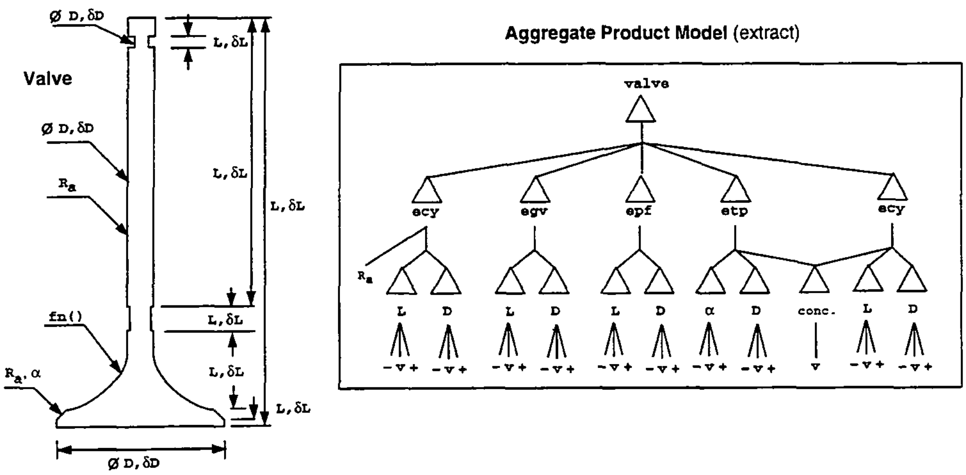


Figure 4.4: Detailed model of the valve component

4.2.3 Dimensioning and tolerancing

In order to support the modelling of product data throughout the product development cycle, particularly in the early stages of conceptual and embodiment design, it is necessary for the model to be able to represent incomplete data in a coherent manner. The model must provide useful information to the manufacturability analysis functions within CESS. The dimensional and tolerance information which is available at each stage of the design process varies. There is a gradual introduction of product detail throughout the development. This is reflected in the number and type of product model objects which can be created. This changing model affects the types of manufacturability analysis which can be performed. In developing the aggregate process planning methodology, the processing possible for different feature design states has been assessed.

In Table 4.2, the results of this assessment are shown. At each stage of development, the possible combinations of feature detail are identified. Against these states is plotted a breakdown of the analysis functions which may or may not be performed. When a feature is added, the dimensions and tolerances will either be specified or left indeterminate. When designing sub-components for a major product, the size envelope and some dimensions of some key features will be known from the start. Other features may not be dimensioned until later in the design process. There are two different methods of specifying tolerances, both of which should be supported in an aggregate product model. Tolerance information may be represented either specifically, using the boundary limits, or implicitly, using standard tolerance intervals (IT). The standard interval of tolerance number defines the variation in dimensions allowed for a given nominal value range.

The product model must allow the design to be in each of these states. Indeed, the different features on a component should be permitted to be in different states at the same time, since the more important features will be detailed whilst the lesser features are at the concept stage. The dimensioning and tolerancing issue therefore presents problems in product modelling. The model must be flexible so that it can represent a feature in any of the states from the table below. Also, the current state of the features in the model should be easily identified.

Table 4.2: The staged dimensioning and tolerancing of designs

Design stage	Dimension & Tolerance: D	IT	Tol.	Manufacturability Evaluation: gps	res.	cps	m/c	t	c	q
Concept	?	?	?	✔	✔	✗	✗	✗	✗	✗
↓	+	?	?	✔	✔	✗	✔*	✗	✗	✗
Continuing development	✚	?	?	✔	✔	✗	✔*	✔*	✔*	✗
OR ↓	?	✚	?	✔	✔	✔*	✗	✗	✗	✔*
Embodiment	✚	✚	?	✔	✔	✔*	✔	✔*	✔*	✔*
Detail	✚	<	✚	✔	✔	✔	✔	✔	✔	✔

key:

?	Parameter values not determined
+	Some parameter values set
✚	Key parameter values set
<	Parameter value defined by implication

✗	Evaluation not possible
✔	Full evaluation possible
✔*	Aggregate evaluation possible

D	Dimension values set
IT	Standard Interval of Tolerance grade
Tol.	Explicit tolerance values on dimensions
gps	Generic process alternatives
cps	Constrained process selection, including consideration of quality capability
t	Processing time calculation
q	Process quality prediction
c	Processing cost calculation
res.	Manufacturing resource, i.e. machine tool types suitable
m/c	Specific machine tools selected

When the feature is first identified, the actual values of the dimensions may not have been determined. In addition, it is unlikely that any indication about the required quality level of the feature will be available. At this stage, the manufacturability analysis is limited to determining the generic process types which can be used for the feature and the machine types which can perform these processes. These process and machine types selected cannot be checked for constraints such as quality capability. The next piece of information which is usually available is the values of the feature dimensions. If only a limited number of dimensions are available, such as a hole diameter but not the depth, then some specific machine tools can be assessed and the process and machine types further refined (e.g., reaming process is not suitable for large diameter holes). If the full set of dimensions are known for a feature, then it is possible to perform aggregate process time and cost calculations, based on assumptions of generic machine tool parameters, for each possible process. These are only partial assessments, since the use of alternative machine tools will vary the times and costs from these average values. An alternative step is to have the quality level defined for a feature, but the dimension values left undefined. This is a less common situation, but could occur where, for example, an interference fit is required between components, but the size of the joint is not critical. In this case, whilst it is not possible to calculate times and costs, it is possible to apply quality constraints to the list of possible processes and to make judgements about the likely quality for those remaining. With further development, the dimensional values and the quality levels are both available. At this stage it is possible to perform all the assessment functions to some degree. When the quality values are only known as standard IT grades it is not possible to make as full an assessment as for the detailed case, when the tolerances of individual dimensions are specified explicitly. There is therefore a distinction in the detail to which the manufacturability assessment can be made for these cases.

4.3 Implementation of the CESS product model

In this section the implementation of the CESS product model will be discussed with reference to a simple example product. Instances of each of the classes of the model will be presented and the attributes and functionality which is associated with the object class will be outlined. Each product which is modelled in the system is made up of a hierarchy of objects, which are instances of a variety of different classes. Each different class represents an increasing level of information as the tree is traversed from root to leaf nodes. At each level in the tree, the siblings of an object will be of the same generic class, although they may be instances of different specific classes. For example, components are made up of many feature objects which are all instances of specific feature classes within the generic feature super-class. The following sections describe each of the classes of the product model, starting with the assembly level and progressing through to the most detailed information.

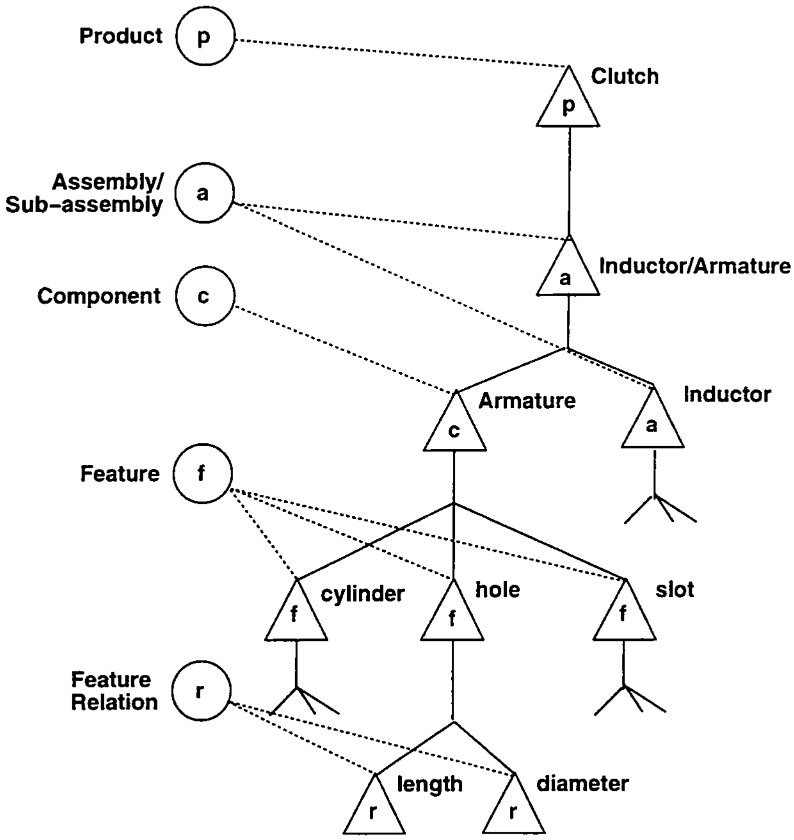


Figure 4.5: A simplified example of the object product model

4.3.1 Assembly class

The assembly class is used to model the combination of multiple physical objects into a single part which can be manipulated and stored as a single item. The creation of an assembly object allows the functions of the system to process the product at the level of its assemblies. In particular, the assembly represents the sum of the components which belong to it. Properties of the assembly class include the size envelope (maximum length, width and height), weight and cost.

The assembly class is divided into a number of different sub-classes which allow the model to represent specific information about different types of assembly. In particular, assemblies may be connected either reversibly or permanently. An example of a reversible connection is a pair of components which are joined with nuts and bolts. Permanent connection methods include adhesives and welding; where assemblies include permanent connections, they are categorised as fabrications, a sub-class of assemblies. This allows the system to provide different functionality for each type of assembly through the creation of fabrication specific methods attached to that class.

Assembly objects can form the root node in the product tree; they can have other assemblies, components and features as child objects. The CESS model supports unlimited levels of sub-assemblies (subject to computer memory availability). Components and sub-assemblies are the most typical children of assembly objects, however, features can be defined at this level. This is particularly common in fabrications, when components are joined together and further machining is then applied which crosses the boundaries between the components.

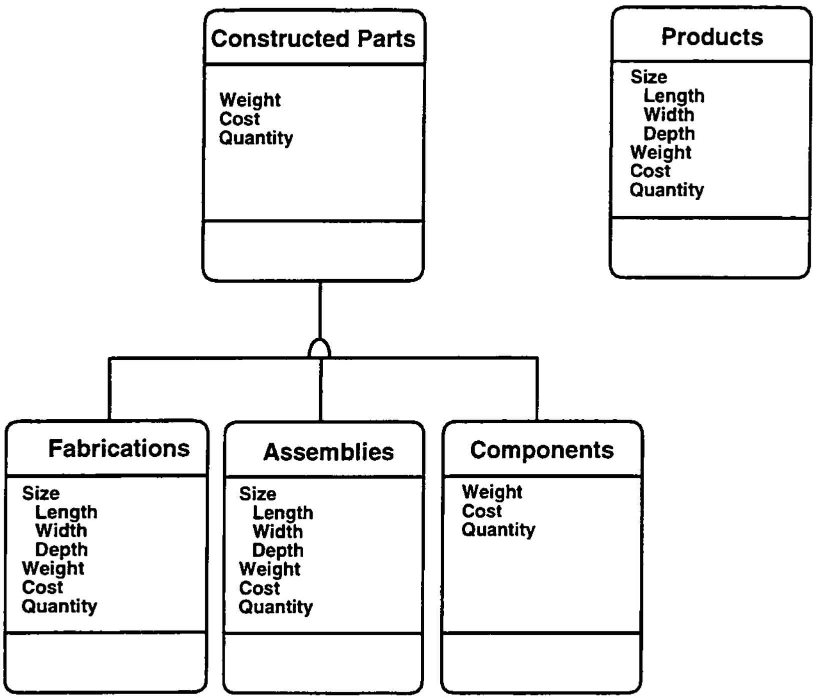


Figure 4.6: Constructed Parts: Products, Fabrications and Assemblies

In addition, an instance of the assembly class can be the "product" which is currently being considered. The product is defined here as the completed part which is sold to the customer. Products can be single discrete components, fabrications or assemblies. The product class allows the definition of additional functionality to the product assembly object.

4.3.2 Component class

A component within the CESS product model is defined as a discrete part which is created from a single piece of material. Components are usually the building blocks of the assemblies. Components can form the root node in the product model tree (where the product is just a single discrete part), but usually belong to assembly objects. In contrast to the assembly model, only a single component class is required, since difference between individual component types are represented at feature level. The component object allows the functionality of the system to address individual parts within the system, about which it stores a variety of attributes: overall geometry, volume, weight, material and cost. The detail of the geometry of each component is stored in the features which belong to it. The component class object only represents the basic information of the part and stores the sum of the properties of the features.

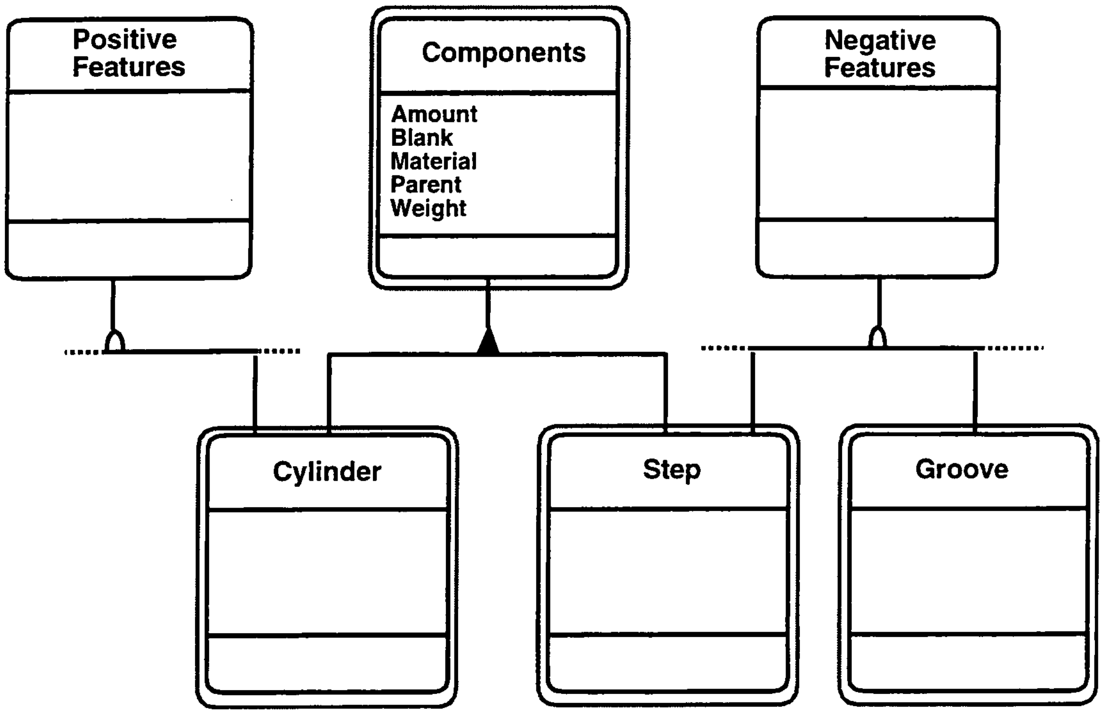


Figure 4.7: Generic component model

The features of a component can be divided into two types: positive and negative. Each component has one positive feature which defines the overall geometry of the part. Examples of positive features are cylinders and prisms. The geometry defined by these positive features is then refined through the addition of negative features. Negative features define material which is removed from the positive feature in order to generate the detailed component shape. These two types of feature are defined using separate class structures, as explained below.

4.3.3 Positive feature class

As previously stated, the positive feature defines the overall shape of the component, or spatial envelope. No part of the component can project outside of the positive feature (with the exception of threads, which can be considered to neither increase nor decrease the initial diameter of the feature upon which they are placed). If projections are to be created on parts, then these are modelled in the CESS model through the definition of fabrications, consisting of more than one component joined together.

Positive feature objects can only exist in the object model tree as child objects of components. Each component has one and only one child positive feature. Positive features have no child objects themselves, they are leaf nodes within the tree. They do, however, have a defined set of attributes. The geometry of the positive feature is defined using attributes (in contrast to that of negative features, see below). Three classes of positive feature are defined in the model at present, as shown below in Figure 4.8.

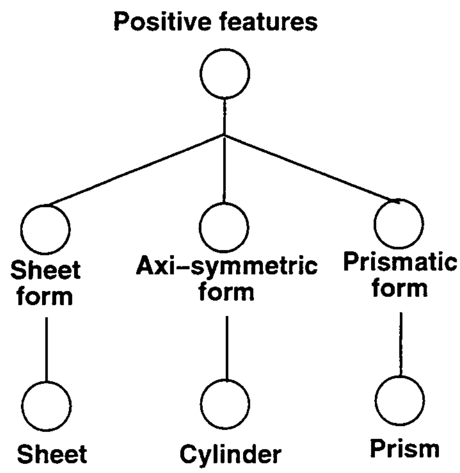


Figure 4.8: Positive feature classification

The figure shows that the positive features all share some basic attributes and methods: Each feature has properties of volume, parent and blank_list. Of these, the volume and parent properties will be defined at run-time for individual instances. The blank_list property is a static property however. This defines the blank types which are suitable for use as raw material for creating components of the shape of the feature. The *Blank* method is used to control the selection of the blank according to the property value. The value of the blank list property is defined for each sub-class of positive feature: components based on cylinders may have blanks which are axi-symmetric, whilst sheet features imply sheet blanks. In addition to those properties and methods defined at the super-class level, each positive feature sub-class has dimensional properties defined, as shown on the figure. The *Dimension* method, inherited from the super-class, is used to assign the values to these properties.

4.3.4 Negative feature class

From the literature review, it can be seen that there is multiple definition of the term the exact definition of a feature, with each researcher redefining the term for the particular task. In the aggregate product model, a negative feature is defined as: *Individual geometric characteristics of a solid part, the sum of which make up the full geometry of the part.*

Negative features belong to a specific member of the negative feature class hierarchy. This hierarchy has been developed to allow the modelling of system functions and engineering decisions for particular geometrical shapes. All feature classes ultimately belong to the *generic features class* which defines the basic attributes and behaviour common to all features. In Figure 4.9, the structure of the generic feature model is shown. Each feature class is constructed from a number of feature relation objects which store the dimensional information of the feature. The figure also shows the properties and methods which are defined for each feature class.

The APM permits features objects to belong to either component or assembly objects. Features belonging to assemblies have the same function as those attached directly to components; but the logical grouping of features at the assembly level allows the designer to specify that the feature should only be generated after the creation of the assembly object, by an assembly or fabrication process. This behaviour enhances the flexibility of the model and allows the swift definition of features which cut through more than one component part.

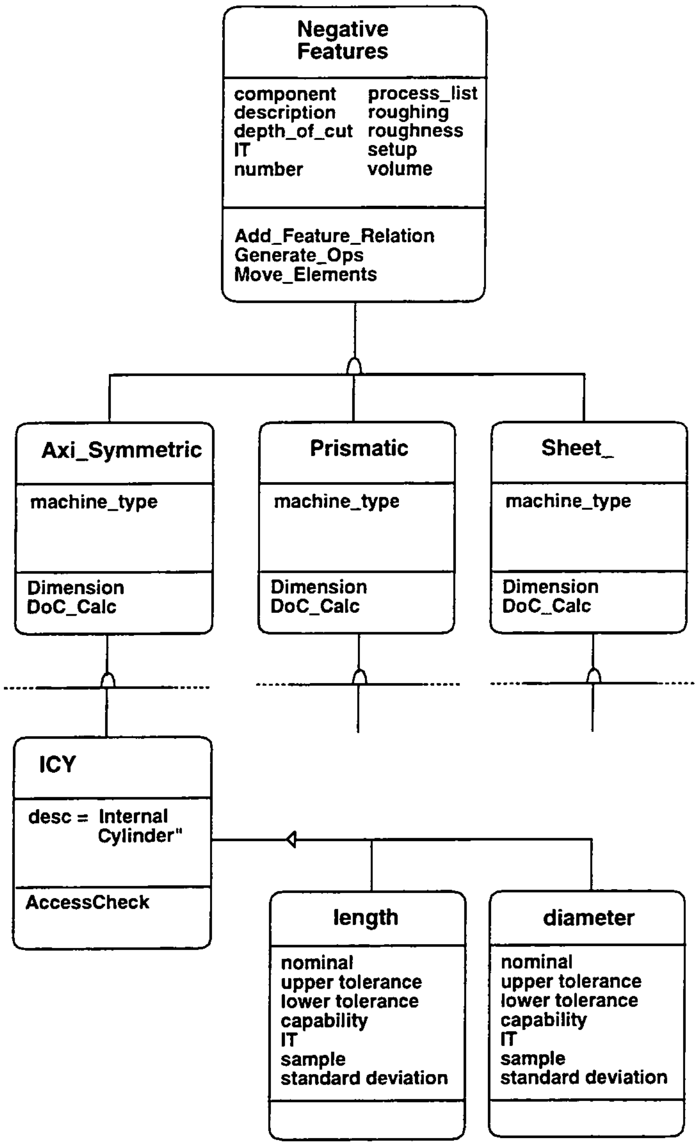


Figure 4.9: Generic feature class

The approach to the definition of feature geometry is the major difference between this model and most feature based models. The feature classes are not defined with a fixed and limited set of geometry information which must be specified in order to store the feature within the model. Instead, the system seeks to allow the user as much flexibility as possible in the definition of the geometry. This leads to the adoption of a two-layered data model for the representation of features. An additional class of objects is defined, called feature relations, which allow the geometry to be stored and modified.

The *Feature Relations* have been proposed as a means of solving the problem of tolerancing and assembly modelling in solid models. Feature relations are an integrated schema for the representation of feature characteristics, including linear and geometrical dimensions and tolerances and also component connections within assemblies. A classification of feature relations has been developed for dimensioning and tolerancing in accordance with the industry standards (BS308:3). Feature relations are discussed in detail in a following section (section 4.3.5).

4.3.5 Feature relations Class

Feature relations are an integrated method for dimension, tolerance and connectivity definition. Feature relation objects can be added as child objects to features in order to specify additional detail about the product geometry. Feature relation objects belong to one of the leaf nodes in the feature relation hierarchy (Figure 4.10). In particular, the APM uses a classification of *geometric* feature relations to store the dimensions and tolerances of all negative features. The classification of assembly feature relations is used to define the way in which individual components are joined together to create assemblies and fabrications.

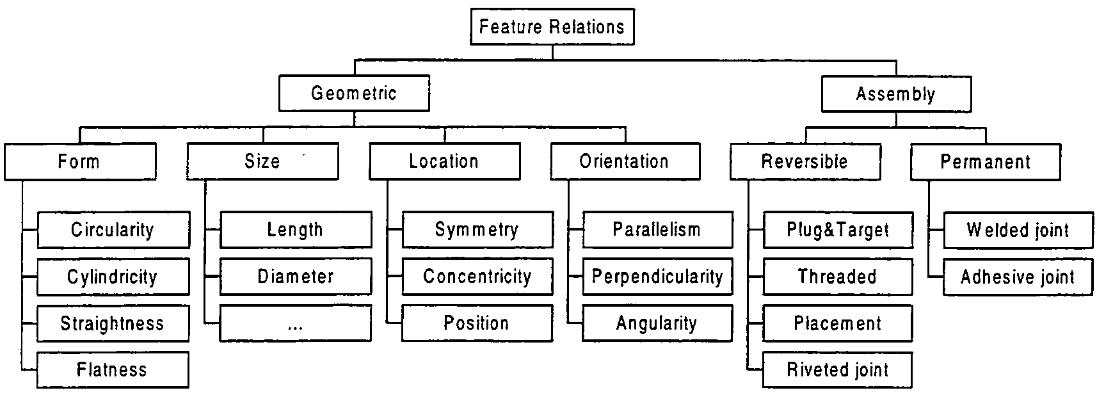


Figure 4.10: Feature relation classification

The separation of the geometrical definitions of feature into sub-objects gives the model the flexibility to support a reduced information set. Additional geometric information can be added to a basic set in order to specify feature details when required, without the need to store these details in the general case.

4.3.5.1 Geometry feature relations

The geometric feature relation class has been introduced as a unified means of representing dimensioning and tolerancing information. This methodology is in accordance with the industry standards for the representation of dimensional and tolerancing information (BS308:3). The geometry feature relation class is subdivided into a classification based on the tolerancing standards. In the product model, the geometry feature relation objects are used to represent both simple geometry such as length and diameter, as well as more complex specifications such as concentricity and flatness. This is in contrast to standard solid modelling methods, where the dimensions of features are defined as attributes of the feature objects instead of objects in their own right. This traditional approach leads to a dichotomy in the representation of "normal" tolerances on dimensions such as length and "geometric", or "form", tolerances such as concentricity. The feature relation approach, however, uses the same representation schema for both types of dimension.

4.3.5.2 Assembly feature relations

The assembly feature relations are used to show the way in which components are linked together to form assemblies and fabrications. These relations are categorised according to the type of connection, initially either reversible (e.g. nut and bolt) or permanent (e.g. weld). The assembly feature relation object represents the physical link between the two components, not the process of joining them. I.e. a weld feature relation represents the physical weld joint and does not store information regarding the welding process, which could be one of several available. The development of assembly assessment functionality falls outside the scope of this project and therefore the assembly feature relations will not be further detailed within this thesis.

4.3.6 Compound feature class

The compound feature class of objects may be used to define a particular set of features which form a standard configuration. These standard configurations may be selected by the designer through the selection of the compound feature, instead of through selecting each feature in turn. A compound feature is made up of a number of feature objects. The features which make up the compound feature are linked to the component in the same way as other features. The compound feature object is another parent of these objects, representing the designer's intent to link these features. This structure allows the system functionality to be the same for features which belong to compound features and those which are independent.

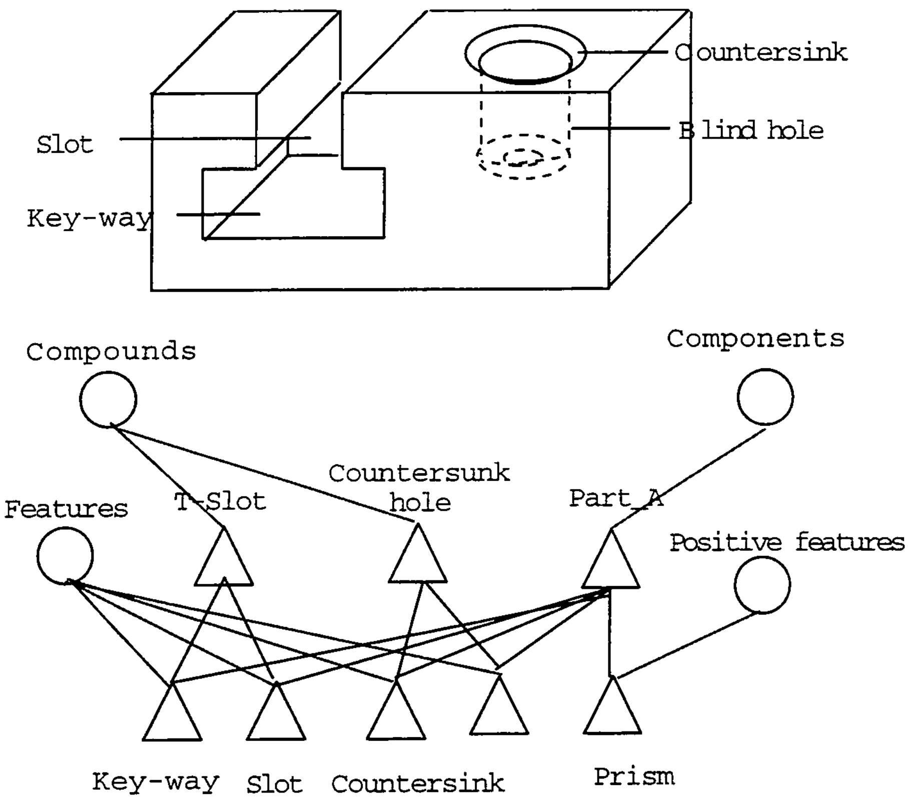


Figure 4.11: Example compound features

Compound features are defined as a labour saving device for the designer and aid process planning, reducing the computational load on the system. Where standard sets of features are specified, these can be represented as compounds, they need not go through the full processing procedure each time to calculate their effects. An example might be a set of holes which are tapped and countersunk: three features are represented as a single compound. Compound features can have the same parent objects as simple features, i.e. components and assemblies. In Figure 4.11 the compound feature concept is illustrated with a simple component, labelled Part A, which has two compound features, a T-slot and a countersunk hole. In each case, the component object is a direct parent of the features, with the compound feature objects also parents of their respective features.

4.4 CESS feature classification

This section will detail the feature set which is used in CESS. This feature set is designed to represent the product through fundamental building blocks. The class of features can be divided into many sub-classes based on the characteristics of the individual features, (Figure 4.12). These sub-classes overlap, as shown in Figure 4.13. It will be seen that there are nine different groups of features which are allowed, which are represented with seven distinct classes.

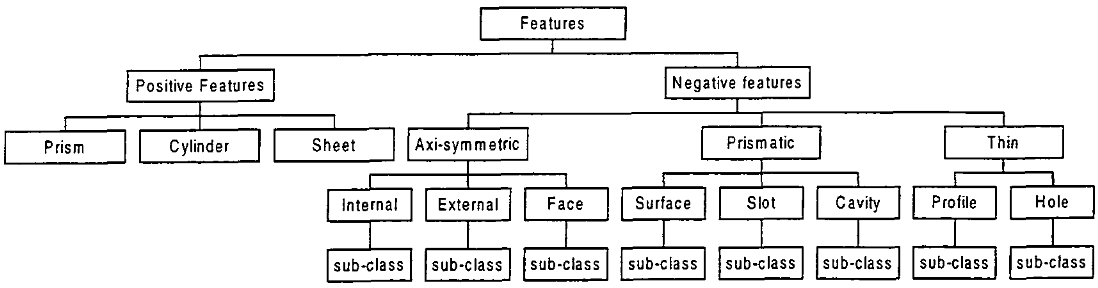


Figure 4.12: CESS feature taxonomy

The geometry of individual components is constructed by the addition of negative features to a single positive feature which describes the basic shape of the component. Since this system is an aggregate model, the full details of any component may not be of relevance to the system. Therefore, the means has been determined to deal with either exact geometries, where all features are fully described, or with indeterminate geometries which have only certain known features and additional unspecified features.

Multiple inheritance is used to allow some feature classes to share the attributes and functionality of two features branches. A typical example is shown in Figure 4.13, where the external ring feature is a child class of both the external and face feature super-classes. The diagram shows schematically that this allows the child class to inherit the properties and methods of both the external and face classes, but not those belonging to the internal class.

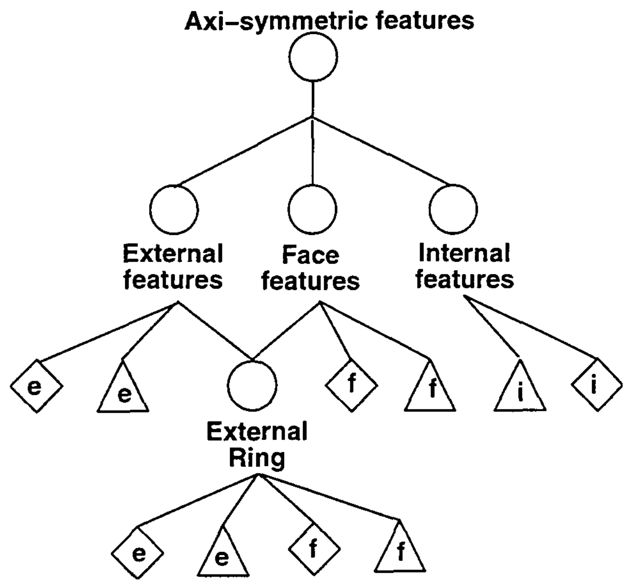


Figure 4.13: Example of feature multiple inheritance

4.4.1 Positive feature classification

Positive features can be of three basic types: sheet, axi-symmetric and prismatic. For each of these cases it is possible to define a additional sub-class which is a special case of the main class, as shown previously in Figure 4.8. Sheet forms can be of any shape, but a *sheet* feature is defined as a rectangular sheet; axi-symmetrical forms may have varying diameters, but the *cylinder* feature has a constant diameter and prismatic forms can be any shape, but the *prism* feature is a cuboid. The general classes are required to represent those components for which it is known that some shaping will be done outside the company and do not, therefore, need to be modelled for the purposes of aggregate process planning. Examples include castings and forgings, which can be modelled as prismatic forms.

4.4.2 Negative feature classification

4.4.2.1 The relationship between feature shape and component shape

The features can also be classified by the basic shape of the host component. There are three of these, prismatic, axi-symmetric and sheet. This classification reflects the ability of different processes to make these features. It is important to note that this class refers primarily to the shape of the component, as defined by its positive feature. This means that for some features which are of basically the same shape, there are two feature classes defined in the product model. For example, a hole of the same geometry might be made in a prismatic component or along the axis of symmetry of an axi-symmetrical component. In the CESS product model these would be represented by objects of different classes: The former would be a prismatic hole feature, the latter an axi-symmetric internal feature. In the case of the prismatic hole, processes which rely on work rotation would not be suitable, whereas they would be preferred for the axi-symmetric hole. It should further be noted that a prismatic hole can be specified for a cylindrical component, this representing any hole which does not run along the axis of symmetry of the part.

This classification is used to restrict the addition of certain types of features to components which are based on positive features from each classification. In other words, prismatic components must not have axi-symmetric features defined, since these are incompatible by definition. Additionally, sheet features and prismatic features cannot be mixed, as they form exclusive sets.

- Prismatic Features

These features have no axis of rotational symmetry. These features may be added to either axi-symmetric or prismatic components. An example prismatic feature is the pocket.

- Axi-Symmetric Features

All these features have an axis of symmetry, which means they can be produced on lathe type machines. Axi-symmetric features may only be added to axi-symmetric components. Components which are based on an axi-symmetric positive feature may have prismatic features specified, but they may not have sheet features specified. A typical axi-symmetric feature is the external cylindrical surface.

- Sheet Features

All these features relate to thin materials, which can effectively be modelled in two dimensions. This allows the use of 2-D processes such as stamping and laser cutting. In addition, this class of features includes the set of bends. Sheet features can only be added to sheet components. Components based on sheet positive features must not have either prismatic or axi-symmetric features specified.

4.4.3 Internal and external features

4.4.3.1 Internal features

All of these features are enclosed by some of the material of the component. This means that the manufacturing process employed has to take into account any access difficulties which this implies.

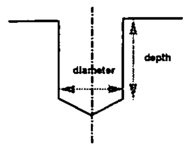
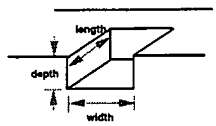
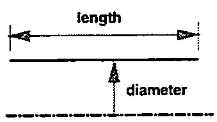
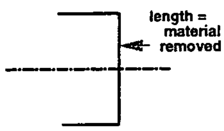
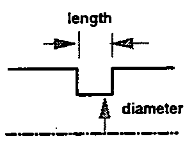
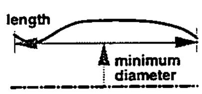
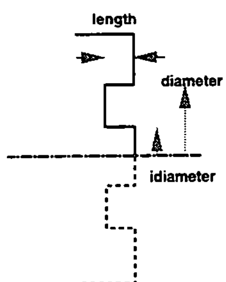
4.4.3.2 External features

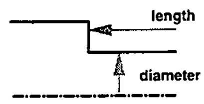
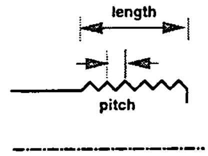
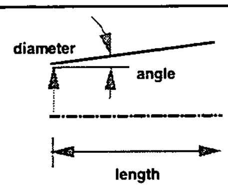
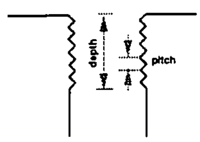
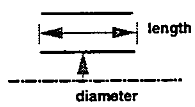
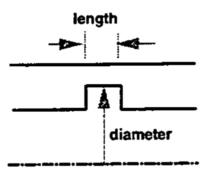
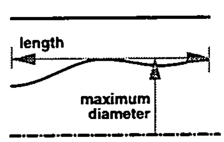
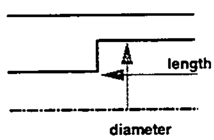
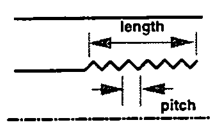
These features are not enclosed by the material of the component but are found on the outer surface. There is no special requirement for access for the manufacturing process.

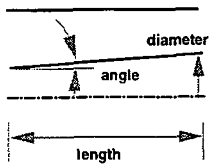
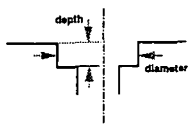
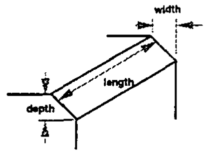
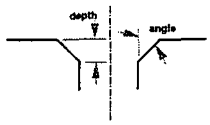
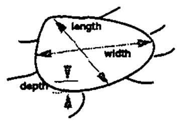
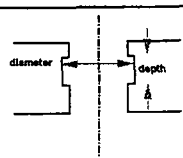
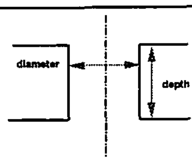
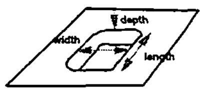
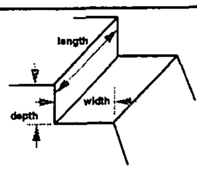
4.4.4 Feature set

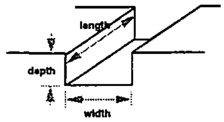
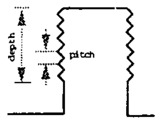
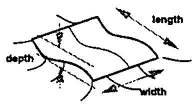
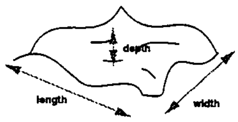
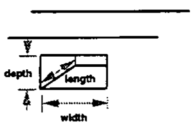
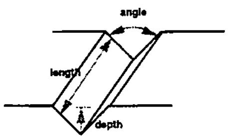
The full classification of features which has been developed for use in CESS is shown in Table 4.3. The first column gives the three character identification code for the feature and a description. The parent classes entry demonstrates the use of multiple inheritance to model the varying geometry. Feature classes belong to up to three super-classes from which they inherit behavioural methods relating to different aspects of the geometry. The minimum feature relations entry shows the default dimensioning data that the system requires to assess the features, whilst the optional feature relations entry indicates the extra information which may be specified for a given feature if required due to a particular geometric shape. In the case of features with irregular profiles, the geometry is represented in a simplified manner. An external profile (epf) feature represents a curved length on an axi-symmetric part. In this case, the model stores the minimum diameter of the feature instead of representing the diameter at every point along the length. Using the minimum diameter it is possible to determine the maximum depth of cut for machining, which determines the process selection. The processing time estimations from features of this kind result in overestimating the material removal. It is acceptable for use in aggregate process planning, however.

Table 4.3: CESS feature set details

Feature Class (code)	Feature Class (description)	Diagram	Parent classes	Minimum Feature Relations	Optional Feature Relations
bho	blind hole		prismatic, holes	diameter, length	
cst	closed slot		prismatic. slots	length, width, depth	radius, angle
ecy	external cylindrical surface		axi-symmetric, external	length, diameter	
efa	end face on a cylindrical part		axi-symmetric, external, face	length, diameter, internal diameter	
egv	external groove on a cylindrical part		axi-symmetric, external,	length, diameter	
epf	external profile on a cylindrical shape		axi-symmetric, external,	length, minimum diameter	
erg	circular groove on the face of a cylindrical part		axi-symmetric, external, face	length, diameter, internal diameter	

esp	external step on a cylindrical part		axi-symmetric, external,	length, diameter	
etd	external thread on a cylinder		axi-symmetric, external,	length, diameter, pitch	
etp	external taper on a cylinder		axi-symmetric, external,	length, diameter, angle	
htd	thread on a non-axial hole		prismatic, hole	length, diameter, pitch	
icy	internal cylindrical surface on a cylindrical part		axi-symmetric, internal,	length, diameter	
igv	internal groove on a cylindrical part		axi-symmetric, internal,	length, diameter	
ipf	axi symmetrical internal profile		axi-symmetric, internal,	length, maximum diameter	
isp	internal cylindrical step		axi-symmetric, internal,	length, diameter	
itd	axi symmetrical internal thread		axi-symmetric, internal,	length, diameter, pitch	

itp	axi symmetrical internal taper		axi-symmetric, internal,	length, diameter, angle	
pcb	counterbore: a square depression around a hole		prismatic, hole	length, diameter	radius
pcf	prismatic chamfer		prismatic, face	length, width angle	
pcs	countersink: a chamfer around a hole		prismatic, hole	length, angle	
pfa	prismatic face: any flat surface		prismatic, face	length, width, depth	
pgv	cylindrical groove in a hole		prismatic, hole	length, diameter	
pho	through hole		prismatic, hole	length, diameter	
ppk	pocket		prismatic, hole	length, width, depth	
psd	shoulder on a prismatic part		prismatic, face	length, width, depth	

pst	slot		prismatic, slot	length, width, depth	radius, angle
ptd	thread on a cylindrical section of a prismatic part		prismatic, face	length, diameter, pitch	
sf2	prismatic curved surface with fixed profile		prismatic, face	length, width, depth	minimum radius
sf3	prismatic curved surface		prismatic, face	length, width, depth	minimum radius
pky	keyway		prismatic, slot	length, width, depth	
vst	v-slot		prismatic, slot	length, angle, depth	

4.5 Dimensioning and tolerancing: Geometry feature relations

The requirements for dimensioning and tolerancing are discussed in this section, together with the way in which these requirements are met by the model. It has already been stated that the means of representing dimensions and tolerances within CESS is via the attachment of geometry feature relation objects to the feature objects of the component, as shown in Figure 4.14. Each feature will have a minimum set of geometry feature relations which are required to define its basic geometry, such as length and diameter. These are always instances of the same geometry feature relation sub-class. Additional geometrical constraints may be specified by the user by adding objects belonging to the other sub-classes.

4.5.1 Geometry feature relation attributes

Objects of the geometry feature relation class inherit three principle attributes: Nominal value, upper tolerance and lower tolerance. Thus, it is possible to specify simply the desired nominal value of a dimension, or to add the detailed tolerance information to it. Tolerance intervals for dimensions are specified using the amount by which they may vary above and below the desired value of the dimension, in the same unit as the dimension. It was decided to adopt this three value notation rather than the alternative of recording only the upper and lower values of the dimension to allow easy processing where no tolerance information was required. Where a lower boundary would make no sense, for example in geometrical tolerances such as cylindricity, this figure is set automatically to zero by inheritance from the sub-class in question.

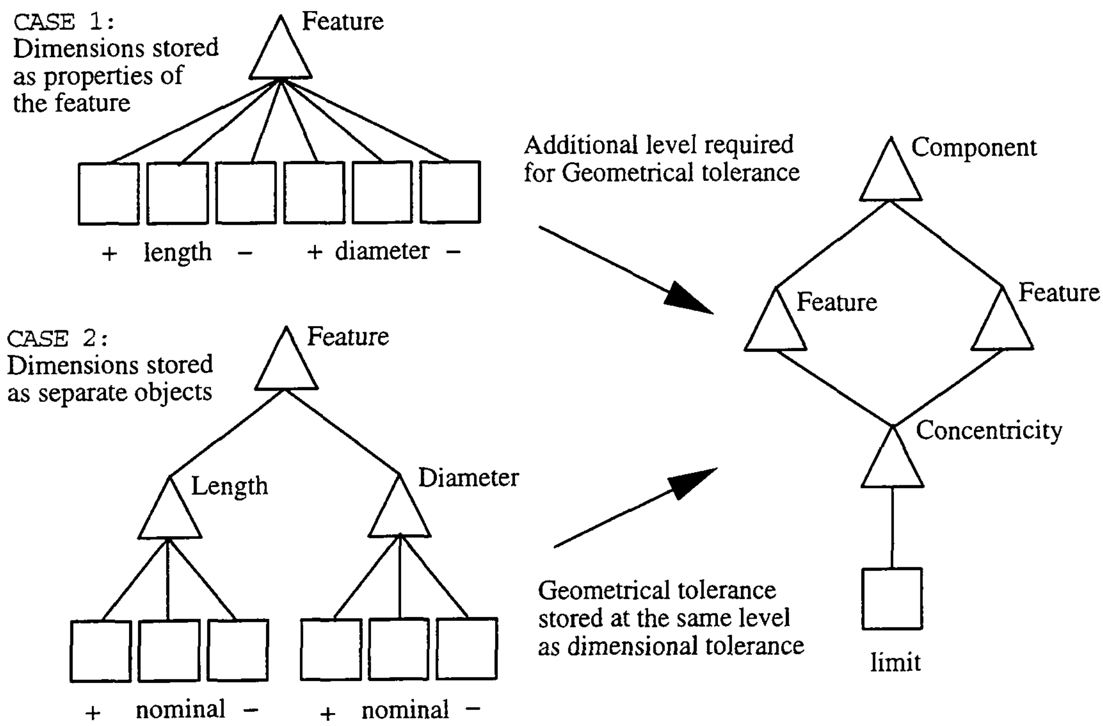


Figure 4.14: Comparison of models for tolerancing

The issue of precedence between tolerancing information is also considered. As set out in the table, the exact upper and lower bounds on a given parameter are considered to take precedence over a standard interval of tolerance setting. The system allows the designer to initially enter an IT value, then revise the dimensioning by supplying exact values of the tolerances. This causes the system to re-evaluate the IT value based on the tolerances and nominal value. The feature relation is a member of a corresponding dimensioning state class. This approach allows the feature relation to inherit the functionality appropriate to its current state of dimensioning. Each state class has methods which are carried out when there is a modification in the dimensioning information, and which will select the appropriate new state class to which the feature relation should be attached.

4.5.2 Minimum feature geometry

The minimum feature geometry depends on the class to which the feature belongs. For example, a cylinder feature has a minimum geometry of length and diameter and therefore needs two geometry feature relations, one for each of these dimensions. The user must specify the nominal values for these dimensions and can if desired specify tolerance values as well.

4.5.3 Additional feature geometry

The requirement for additional feature geometry will occur when a designer wishes to specify a particular geometrical tolerance on a feature. This addition will generally link together two features by specifying that the geometry of one feature is dependent on that of another. One feature is defined as the datum feature; the second one must be manufactured to a geometry relative to that feature. Geometrical tolerances can be used to constrain a large number of aspects, as was shown in the classification in the previous figure.

Of particular interest to the designer using CESS will be the specification of positional constraints. This allows the designer to represent requirements for features to be placed at specific distances apart and give tolerances to these values. The positional tolerance between two features is often of critical importance to a design, for example where two features locate another component which must form an interference fit with the main component. By adding a geometry feature relation manually the designer can force the manufacturability assessment stage of the system to take this quality constraint into account; the system can then return data on the likely cost of this design feature.

4.5.4 Standard tolerances

Since the designer is allowed to leave tolerances unspecified if they have not been determined, the assessment modules are required to make assumptions of the likely values of these properties. The system is able to apply a standard tolerance level to any untoleranced dimensions using the IT value of tolerance. With an IT value and the nominal dimension, it is possible to determine the correct tolerance interval. The system allows the designer to apply a single IT value at any node of the product model and all objects below this will inherit this value, unless there is a tolerance range specified explicitly for that node.

4.5.5 Routing properties

Feature relations require additional properties which are used to store information generated during the routing procedure. These properties are not strictly part of the product model, since they relate to the temporary data of routing and process planning. However, it is convenient from a programming point of view to attach these properties to already existing objects rather than creating unnecessary additional classes and objects.

4.6 Assemblies: Assembly feature relations

The Assembly feature relations provide the system with the ability to model mechanism designs and allow the aggregate assessment of assembly production and DFA. Assembly connections in mechanisms are determined early in design, often prior to dimensioning, so it is fitting that the APM can represent the assembly configuration of these parts. An assembly feature relation defines features on two or more distinct components which are linked together by a joint relationship such as a threaded joint (e.g. nut and bolt) or a placement joint (e.g. hole and boss). The assembly feature relation is a child object of the assembly object. The features which are linked are specified as a property of the object. A classification of assembly feature relations can be developed to represent all the methods of joining parts together (Figure 4.15). The assembly feature relations represent the joint which is created rather than the process of creating that joint. Thus, a weld joint represented as an assembly feature relation could be produced by a number of alternative welding processes.

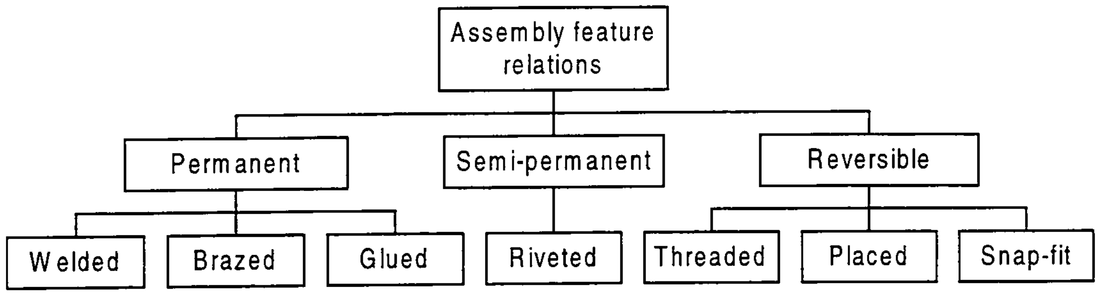


Figure 4.15: Assembly feature relation taxonomy

4.7 Example Product Model

This section will detail the use of the CESS product model for the representation of the design of a real product. It will highlight examples of objects from each of the product model classes in order to demonstrate the full functionality of the product model. The product chosen is an electro-mechanical clutch, which consists of a number of sub assemblies. The basic shape of the product is axi-symmetric, as it is a rotating mechanism, therefore the majority of the parts are based on axi-symmetric features.

4.7.1.1 Top level product model

The product has five child objects, of which three represent assemblies, and two represent components (Figure 4.16). The CESS product model allows any number of assemblies and components to be attached to the product. In this example, there are many levels of assemblies and sub-assemblies belonging to the product (multi-level bill of materials); in other products there might be no assemblies apart from that of the product itself (single level bill of materials). Each of the child objects of the product will have child objects of its own which would define the product in further detail.

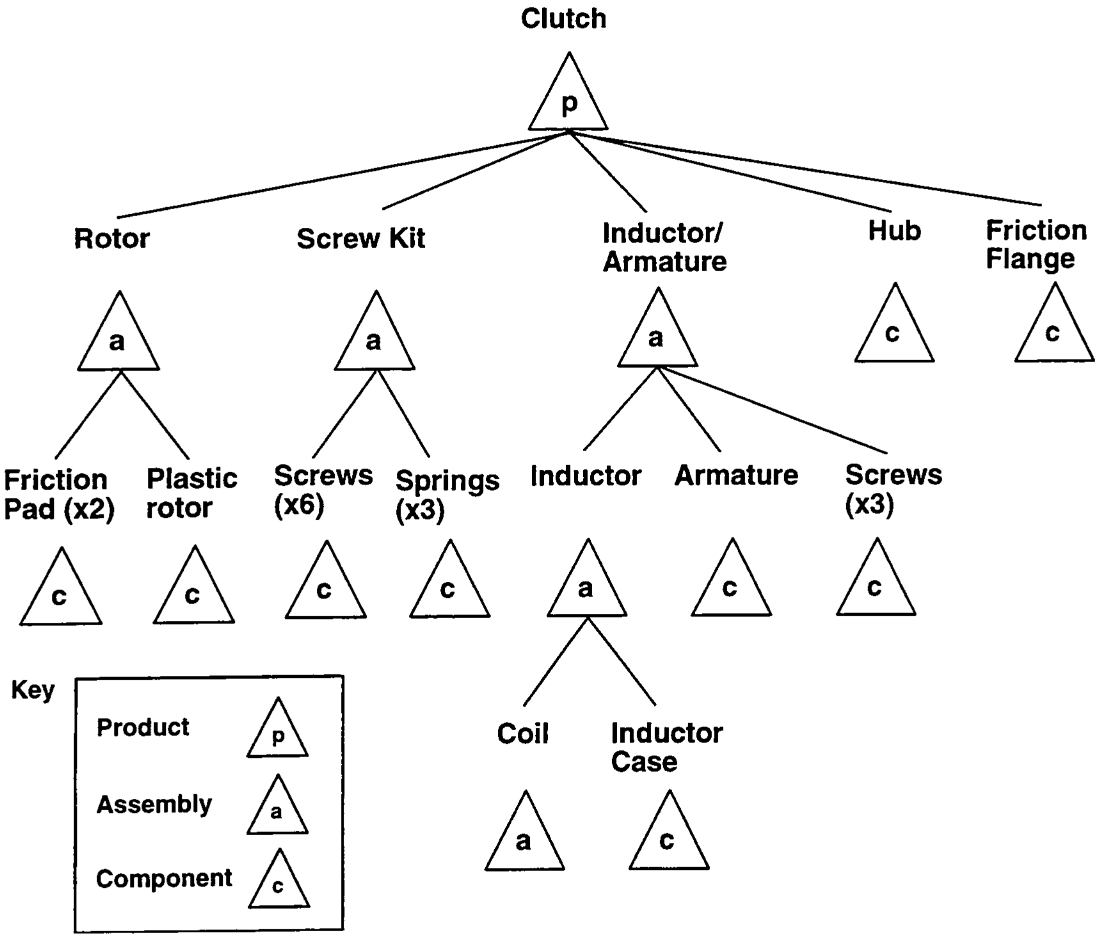


Figure 4.16: Example product model: top level

4.7.1.2 Assemblies and components

Figure 4.17 focuses on the details of a particular assembly. This highlights the flexibility of the CESS product model in representing multiple levels of assemblies and types of part. In this figure it can be seen that the system does not require models for each part, only for those which are to be assessed for machining manufacturability. Thus, the coil assembly, which is manufactured using coil winding processes, is not modelled. In addition, simple standard parts such as the screws, which will not be manufactured in the factory are also not modelled in detail, merely represented as costs.

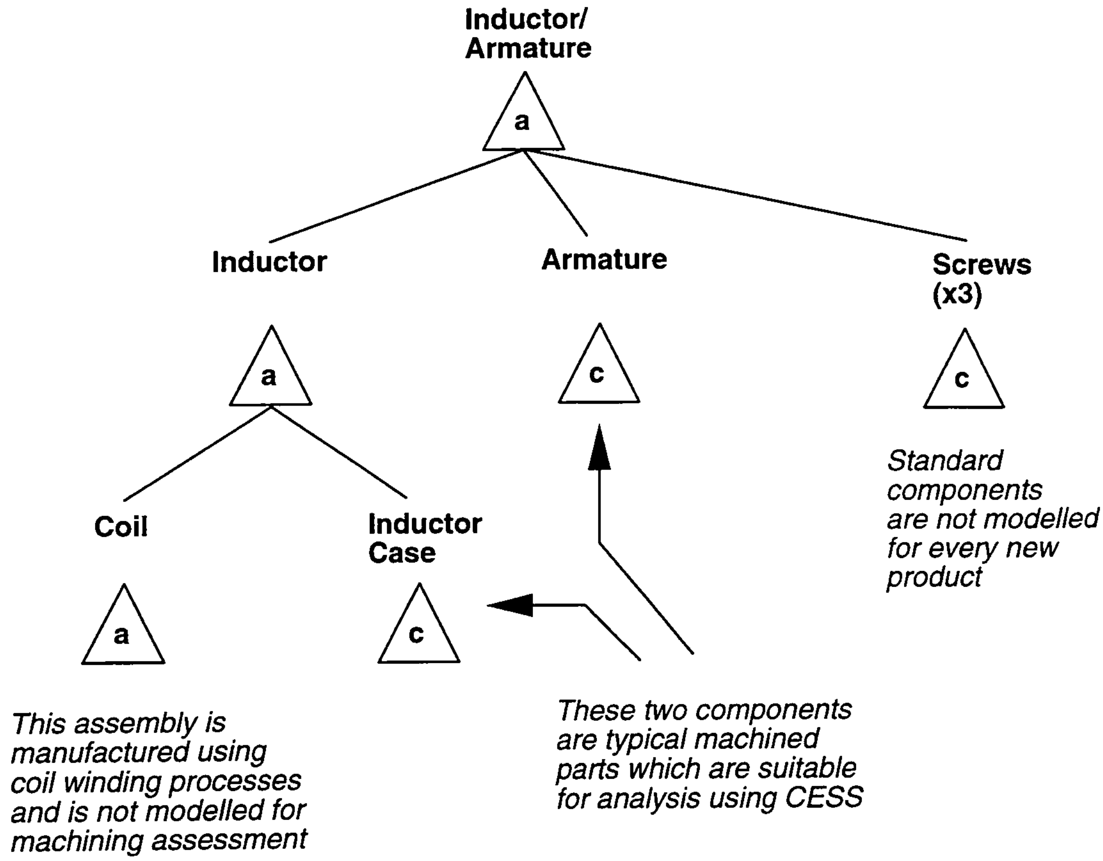


Figure 4.17: Example product model: Inductor/Armature assembly

The figure shows two of the components attached directly to the product: The hub component and the plastic assembly. The hub is a single hollow cylindrical part with a bevelled exterior. The rotor assembly consists of two discs of friction material (the lining) which are to be glued to either side of an axi-symmetric plastic moulding (the carrier). This moulding is a complex shape which fits the geared teeth of the hub. For the purposes of aggregate planning, however, the exact shape of the moulding is immaterial since it will not affect production times greatly; therefore it is defined simply as an axi-symmetric form feature. The lining components are modelled as sheet form features. In this case the designer knows that it is necessary to machine the assembly after gluing in order to ensure a smooth outer surface, therefore a feature has been added at the assembly level to force the aggregate planning to account for this. This is an axi-symmetric external feature.

The two components highlighted in Figure 4.17 as suitable for analysis using CESS are shown in Figure 4.20. Of these, the details of the model for the armature is shown in Figure 4.18.

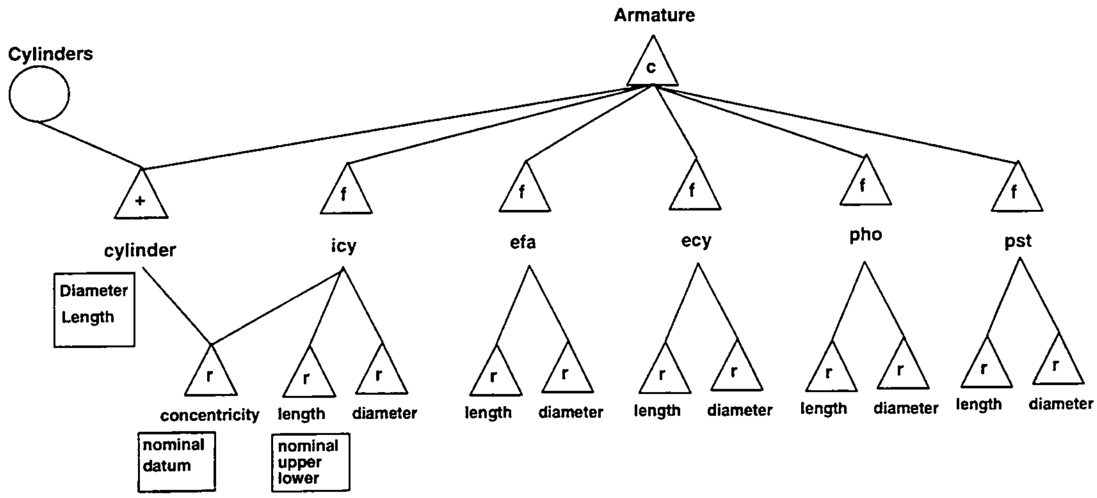


Figure 4.18: Example component model

4.7.1.3 Minimum feature geometry

The feature relations attached to the features of the armature component demonstrate the concept of the minimum feature geometry. Each of the features has two geometry feature relations (from the size sub-class), which are automatically added when the features are created. These provide the minimum geometrical information necessary to perform the aggregate process planning. As soon as the feature is selected, this minimum geometry is implied. At a later design stage, the designer may wish to add more detailed information; this is also shown in Figure 4.18.

4.7.1.4 Assembly feature relation

The plastic assembly also gives an example of an assembly feature relation being used to specify that two components are joined together to form an assembly. In this case the designer wishes to specify that a glued connection is desired: the thinness of the lining and the fact that parts move over both sides, allied to the materials being used, make this the only feasible option. The properties which are specified to the glue feature relation will be used by the assembly aggregate planner to suggest suitable gluing processes. In this case a Redux glue is used for the product at present.

4.7.1.5 Additional feature geometry

The moving armature detail, shown in Figure 4.19, includes examples of the designer specifying additional feature geometry. For clarity, the feature relations specifying the minimum feature geometry have been excluded from this detail. Three geometrical tolerances have been specified with feature relations: The coaxial (or concentricity) feature relation is added by the user to indicate the importance of the hole through the flange remaining concentric with the flange itself. This hole accommodates a rotating component, thus it is important that the hole is not eccentric so that the assembly is free to move. The concentricity object has properties of specifying which parent feature is the datum, and the limit on the tolerance. In this case, the cylinder positive feature is the datum and a concentricity of 1 mm is specified (limit = 1). In addition, two other geometrical tolerances are specified: the holes must be positioned accurately to mate with other holes on another component, and therefore a position feature relation is added to the feature object. Similarly, the slots are required to be symmetrical.

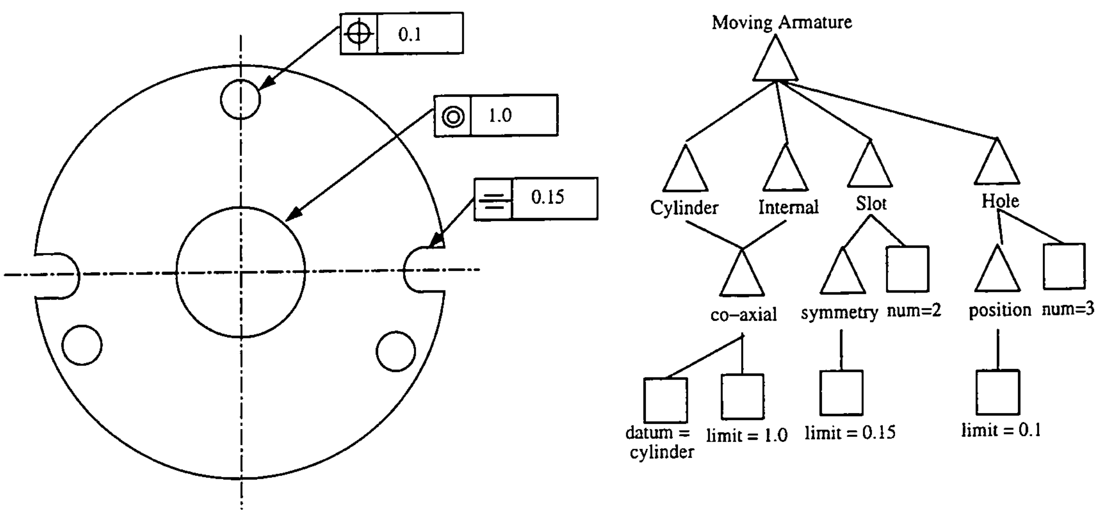


Figure 4.19: Component example showing geometrical feature relations

4.7.1.6 Assemblies and sub-assemblies

The most complex assembly attached to the product is the inductor/armature assembly, named after its two principle components (Figure 4.20). This is modelled as eight components, which are grouped into two sub-assemblies at two further levels.

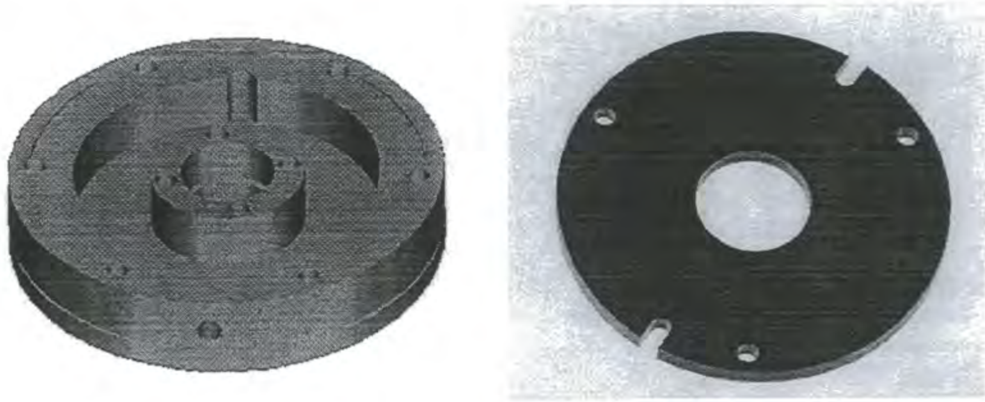


Figure 4.20: Inductor and Armature components

4.7.1.7 Set-up feature relation

Included in the example inductor component is a set-up feature relation. This is an example of the designer wishing to restrict both the way in which the aggregate process planner may order the processes which create the features and the machines on which they may be made. This will generally be made to reflect some additional design or process knowledge. In this case, the designer knows that the features are all very similar and could be done using the same machine. By forcing the aggregate process planner to always keep these features in the same set-up, the designer narrows the system's search space by cutting off known bad routes. These would require extra set-up time and reduce the quality achievable without giving any benefit. Specifying this set-up feature relation may not be necessary; it will however speed up the analysis and save the designer's valuable time.

4.8 Manipulating the product model

This section will discuss the various functions of the system relating to the manipulation of the product model by the user. In addition the requirements for transfer of data between CESS and other systems which might be used in the product development environment is discussed.

4.8.1 Product data exchange issues

This section will discuss the requirements for product data exchange which are to be met in the development of the CESS product model. In order for the system to be integrated into a real engineering design environment it is essential that information may be retrieved automatically from the proprietary design software systems which are employed. It is not the aim of CESS to be a dedicated computer aided design tool in the traditional sense. This task is best performed by the innumerable CAD systems which are currently available. In particular, the functionality of three dimensional CAD systems such as Pro/Engineer, CATIA, Euclid and IDEAS makes them far more suitable for formulating the product design. What is required, therefore, is a means of taking product data from systems such as these and extracting the information required to create the CESS product model.

Since there are many different CAD systems on the market, each using a different format for modelling the product both within the CAD system and in stored data files, it is not feasible for a system such as CESS to be developed to read and manipulate the data models of all available other systems. For this reason the Standard for the Exchange of Product Data (STEP) is being developed by the International Standards Organisation to facilitate the transfer of product models from one system to another by the use of an intermediate common file format, which each system should be able to read and write in. This, theoretically, enables software developers to retain a proprietary modelling system, whilst providing cross system compatibility through a standard file format, avoiding writing a translator for each system with which it might wish to share data. However, because of both the vast number of different uses to which product data is put and the requirement for the STEP standard to be generic, the development of a complete workable standard is not progressing sufficiently to allow widespread use. In most fields of engineering the standard is still in its infancy, with little take up in industry. The exceptions appear to be in the aeronautical and automotive industries, which were the main drivers in the STEP project.

4.9 Conclusions

This chapter described a flexible product model, which can represent design data over the early stages of product development. All the information which is required for aggregate process planning and design assessment can be stored. This includes part geometries, assembly and sub-assembly groupings, component connections via specific features, geometric, dimensional and surface quality requirements and material types. The product model has been designed to be compatible with the emerging STEP standard to enable rapid data transfer from solid modelling computer aided design environments.

With a suitable product model, it is possible to analyse the production options which are available and to make suggestions as to the best production route. Alternatively, the processing information can be used as a design feedback to alter the design in order to produce a product which is either cheaper to make, or can be made to a higher quality. In order to achieve this aim, it is necessary to develop a model of the manufacturing processes and how they are applied. A detailed description of the process model developed for this purpose is given in Chapter 5.

Chapter Five

Process Model

This chapter describes the work relating to the process model of CESS. The process model consists of a hierarchical taxonomy of individual *aggregate process models* together with attendant architecture and functionality to allow the models to be used in concurrent engineering manufacturability assessment. Aggregate process models are simplified descriptions of the capabilities, requirements and parameters of manufacturing processes which allow aggregate process planning to be carried out. The philosophy of aggregate process models is that they should provide the means of making adequate predictions about production with either uncertain or incomplete knowledge. The CESS process model contains a comprehensive hierarchy of manufacturing processes since the approach taken is generic. Due to the limitations of time, however, aggregate process models have been developed for only a selection of processes, specifically the conventional metal cutting (machining) processes. The process modelling architecture also takes account of the need for an aggregate assembly process model, and this is discussed briefly herein.

5.1 Introduction

The machining process model embodies expert manufacturing knowledge which enables CESS to perform automated aggregate process planning and manufacturability evaluation for machined parts. The overall aim of the model is to rapidly generate multiple process and manufacturing plans, using the limited information available during the early product design stages. This is vitally important in order to apply process planning considerations within concurrent engineering. The objectives are the early identification of manufacturing constraints and bottlenecks, and the definition of the best product configuration and manufacturing methods. The most suitable process plans will be maintained and form the basis of fully detailed and optimal plans which will be created using detailed process planning systems.

The aggregate process modelling of manufacturing processes involves the translation of product design data into initial manufacturing planning information. Ideally, those activities should be performed as early in the product development cycle as possible, since then there is a wide choice of options both in terms of product configuration and process selection. Aggregate process models are obtained by the controlled simplification of detailed process models so that they can operate using the limited product information available during the conceptual and embodiment design stages. Any such simplification will almost inevitably result in a loss of accuracy in the associated manufacturing planning predictions. This drawback is outweighed by the ability to rapidly evaluate alternative product configurations and processing options at an early design stage so that the best option can be adopted.

5.2 Principles of Aggregate Process Models

In the development of the aggregate process models for CESS a number of principles have been applied. These principles, which were identified initially by Maropoulos (1995d), were detailed in order to provide a specification for the process models in the system. In this section the principles of process planning are identified and discussed in context.

5.2.1 Controlled simplification of the detailed process models

The complicated process models which are utilised by detailed process planning systems and by process analysis packages are unsuitable for aggregate process planning. These models require too much computation to produce results, and too much data. This information overload will swamp even today's computer systems when applied across a wide selection of features and machine options at the same time. Therefore, it is necessary to simplify the models so that the core function is retained whilst unnecessary processing is not considered. It is important, however, that this process is controlled, so that the information which is retained still provides an accurate assessment of process performance.

5.2.2 *Limited data input requirements*

Of key importance to the aggregate process models is that they require a limited set of data inputs. This is necessary if the models are to be use in the conceptual and embodiment design phases when full product data is not yet determined. It will also allow the system to be written to respond quickly enough to provide design feedback. The aggregate process models should incorporate only the basic geometry information of the component parts and their features, without requiring the specification of more detailed aspects which might not be determined until the detailed design stage.

5.2.3 *Perform core capability checks*

The aggregate process models must retain the ability to check the most important process constraints for features, so that processes are not suggested for features which are clearly impossible from the information available. However, detailed capability checks, that require long computation procedures, and those constraints that will only rarely be broken can be omitted to allow rapid generation of parts. An example of a check which should be made is that boring processes require that a hole is made in advance, whilst full geometric checking for tool path interference is best left to detailed systems such as CNC code generators.

5.2.4 *Model manufacturing operations*

The process models should allow the automated aggregate process planner to model the manufacturing operations as they would be carried out on the shop floor, so that production routes may be passed to the process planning engineers for consideration. Additionally, process plans at the feature and machine tool level allow a contribution to be made to factory layout design and production management.

5.2.5 Measure manufacturing performance

For each process, it is necessary to measure a key set of manufacturing performance indicators. These are the Quality, Cost and Delivery implications of the process. All process models must allow the calculation of cost and delivery through the provision of processing and handling times. Furthermore, quality information must be available for each process.

5.2.6 Utilise company-specific knowledge

The process models must take into consideration the individual characteristics of the particular company's products and process knowledge. Also, the process models must incorporate inputs from the factory resource model so that the processes are evaluated according to the standard of facilities available to the company.

5.2.7 Function-driven operation

The process models should be oriented towards providing the necessary functionality for aggregate process planning. An object oriented approach with encapsulated methods for interacting with the process models is the preferred structure.

5.2.8 Conformance with standards

The process models must conform to all relevant engineering standards, including modelling (i.e. STEP, ISO 10303) and quality management standards. Appropriate levels of cutting data must be used when making assumptions about process parameters, and the suggestions should follow recognised practice.

5.2.9 Conformance with team based engineering

The process models must support the conformance of the overall system to the team based approach to concurrent engineering. This means that the models should be accessible to and usable by developers from all disciplines within the company, and not restricted to use by process planners through over-complexity or the requirement for process planning expertise.

5.3 Machining Process Model Overview

The machining process model consists of a generic classification of manufacturing process types which covers all forms of mechanical material removal. Machining processes are used to produce the finished part geometry, either from the initial raw material form or from partly processed blank workpieces, after primary processes have generated an initial component shape. Machining processes involve the movement of a tool against the workpiece to remove material and they may be categorised in various ways. Machining processes include single point cutting (e.g. turning and boring) and multipoint cutting (e.g. milling and grinding). Machining processes may use linear motion only (e.g. planing), rotational motion only (e.g. cylindrical grinding) or more commonly a combination of linear and rotational motion (e.g. turning and milling). Processes involving rotational movement may be further classified according to whether the majority of the cutting power is supplied through the workpiece or through the tool. Since these categories overlap to some extent, it is possible to define alternative taxonomies. The selected taxonomy will depend on the purpose to which it is being applied.

The machining process model defines methods to determine the suitability of each process class for the manufacture of a given product feature. This assessment is based on feature parameters including material type, dimensions and tolerances. The model is designed to operate using product model data available in the early stages of product design.

5.3.1 Process Class Definition

Each process class within the model must contain a set of parameters common to all classes that are used in the generic process planning functions. In addition, the process classes have a set of methods which define the specific process planning knowledge. The principal method required by a process is the processing time method. The aggregate process planning function selects the processes based on a cost criterion which is derived from the processing time, the setting up time and the transport time. The system must be able to calculate a processing time for each of the process options for any feature, and this is done using the processing time method. This factor has been considered in the design of the CESS product and process models so that a generic method could be devised where possible for each process.

For each process type, a method is used to calculate the processing time for an operation element, which is defined as a single processing step involved in making a feature. These methods take the dimensions of the features, component material definition and machine tool parameters as their inputs.

The geometrical capabilities of processes are not stored directly within the process class, instead being stored as properties of the features of the product model. Each feature has a list of processes which can be used to create it. This property is set using a matrix which relates each process to every feature, indicating whether the process can be used. This is the process to feature capability matrix, reproduced as Appendix A.

5.3.2 Process Classification

The process classification has been developed from a number of literature sources. The primary benefit of classifying processes into a hierarchy, instead of using a simple flat structure is the ability to use the concept of *inheritance* to share functionality amongst similar process models. This reduces the amount of programming to be done and the size of the programs developed. Thus, in the model chosen, all turning processes may be given the same basic set of attributes by defining the super-class of turning, and linking each detailed turning class as child classes.

In the CESS process taxonomy, the machining processes classification has been implemented, and therefore discussion will centre on this branch of the tree. The machining processes have been divided into two branches: single-point cutting and multi-point cutting. This classification is derived from that of Allen and Alting (Allen and Alting, 1985). It differs from that classification, however, in the higher level classification of processes: Whereas Allen and Alting divide the processes first according to tool motions, with the number of cutting points a secondary dividing factor, this classification groups all single-point processes together, separate from all multi-point processes.. The choice of which structure for the process models has been made on the basis of simplifying future programming, since there are good arguments in favour of both approaches: motion oriented categories are suited to models which represent the product as a set of created surfaces, and allow the automatic interpretation of a machine tool's process set from its position in the machine tool hierarchy. On the other hand, the cutting oriented approach groups together processes with similar mechanics of cutting at the detailed level, and similar cutting condition requirements. This results in the process classes having more similar attribute lists in the second approach, and therefore gets the best out of object oriented modelling. It is interesting to note that the classification of the processes is dependent of the classification of machine tools and *vice-versa*: the two classifications should be harmonised to allow efficient representation of the relationships between machines and the processes they can perform.

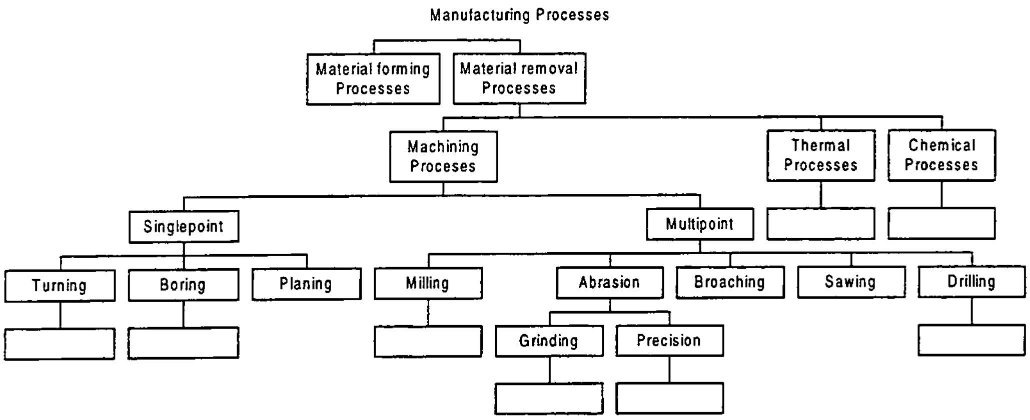


Figure 5.1: High level process taxonomy (after Allen and Alting, 1995)

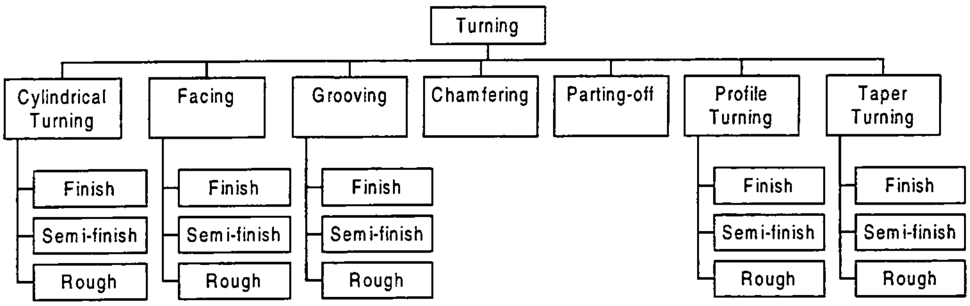


Figure 5.2: Turning process taxonomy

The higher levels of the CESS process taxonomy are shown in Figure 5.1. The blank boxes represent the areas where additional classes are used to model the processes in more detail. Thus, turning and boring are modelled using sub-classes, and therefore several more detailed models, whereas there is just a single process model for the planing process. The turning taxonomy (Figure 5.2) and the boring taxonomy (Figure 5.3) are essentially the same, with the absence of parting-off being the only difference. These processes are the external and internal equivalents of each other.

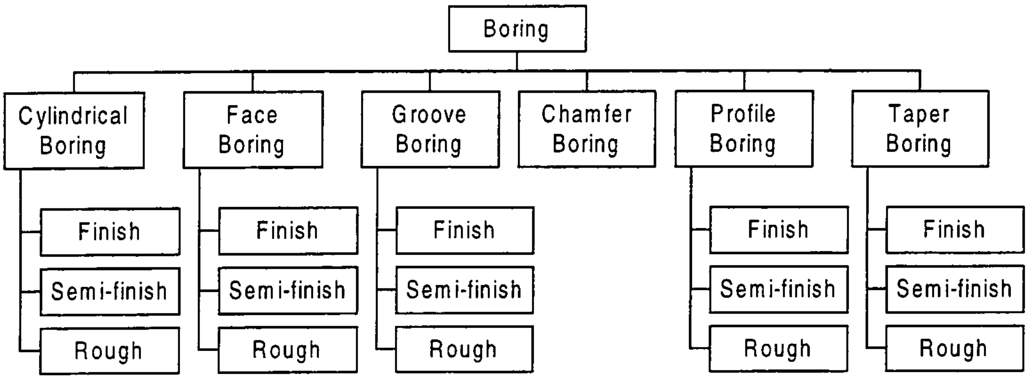


Figure 5.3: Boring process taxonomy

The milling taxonomy (Figure 5.4) reflects the variety in tool shapes which are available for the milling process as well as the disparate geometries which can be created. Thus, multiple milling processes may represent alternative tools for certain simple geometries whereas other processes may represent the generation of a particular more complex geometry (e.g. cavity milling).

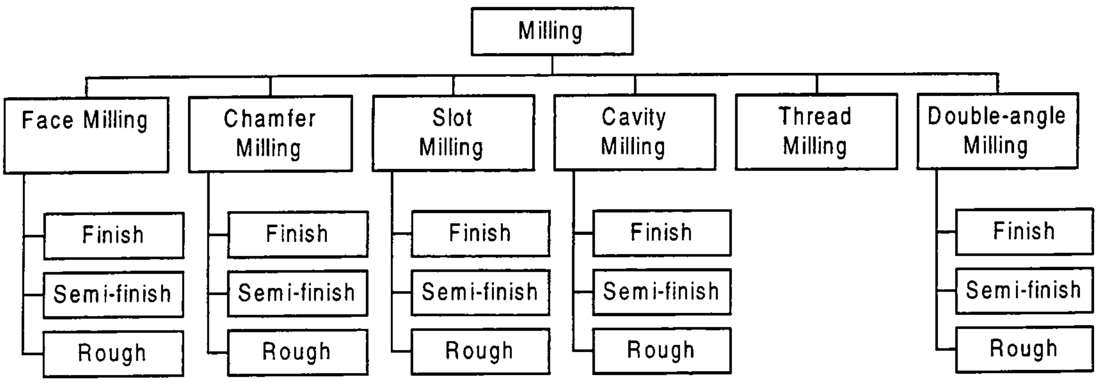


Figure 5.4: Milling process taxonomy

Abrasive processes are divided into grinding (Figure 5.5), which covers the main high volume processes (which may also perform finishing operations), and the precision abrasion processes (e.g. honing and lapping). It is divided into processes that generate primarily cylindrical surfaces, and those which generate only flat surfaces. Cylindrical grinding is further divided according to whether the workpiece is clamped in a centre-chuck arrangement (centre grinding) or not (centre-less grinding). Flat grinding is subdivided according to whether the workpiece is fed parallel to the wheel direction as well as along it (traverse grinding) or not (plunge grinding). Precision processes are specialist methods generating very smooth surfaces and very tight tolerances, and are used only as finishing processes after another process has generated the basic feature geometry.

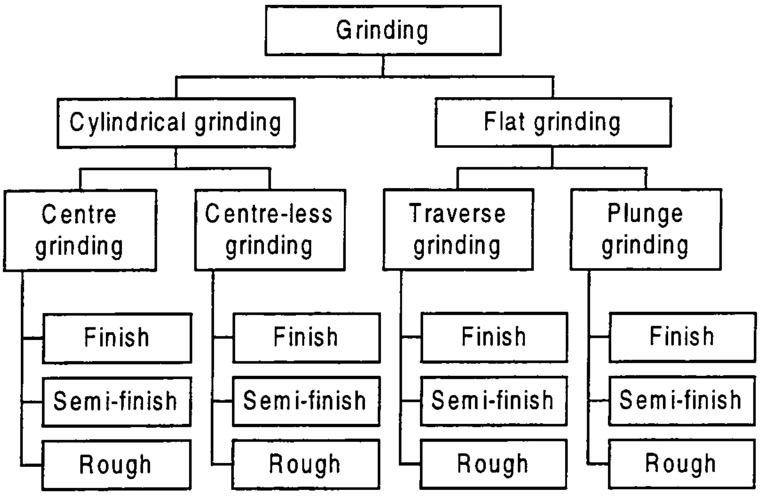


Figure 5.5: Grinding process taxonomy

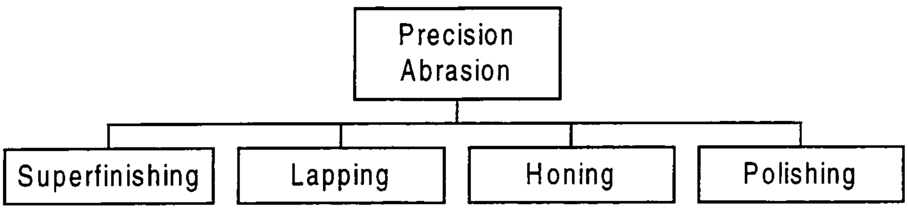


Figure 5.6: Precision abrasion process taxonomy

The drilling process group represents a range of processes which have the same basic cutting motion of a cylindrical tool, rotating with respect to the workpiece, that is moved only along the tool axis. As for milling, the generic drilling can be varied in two ways to produce distinct sub-processes, shown in Figure 5.7. On the one hand, the cutting mechanics may be altered, as in the case of reaming, threading and some types of trepanning. On the other hand, a different tool shape or size may be used to generate an alternative geometry. For example, the chamfering process does not differ in application to drilling, but the tool shape and the initial conditions of the workpiece define it as a separate process requiring its own individual model. Threading and tapping are not strictly drilling processes, but they are carried out on the same machine.

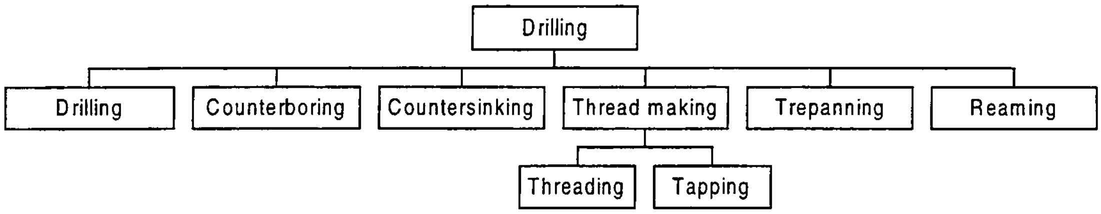


Figure 5.7: Drilling and associated processes taxonomy

In summary, this process classification covers the whole range of machining processes, enabling each process to be modelled according to the generic form determined by the processes super-class. Many of the processes are divided into three sub-classes according to the level of quality which is being generated using them. The exceptions to this rule are processes where the range of output quality from the process is limited, such as drilling, where the process quality is hard to vary, and the fine finishing processes, where the goal is always very high quality and the process is never used for roughing.

5.3.3 Manufacturing operations

In order to build a useful process model, decisions must be made about the level of detail to which the process will be represented. It is quite possible, and indeed common for research into specific process optimisation, to model the process at the physical level, trying to represent the process in terms of the basic laws of physics or even to analyse the behaviour of materials at atomic/molecular levels. However, this is clearly inappropriate for a system which aims to provide rough-cut planning information about a wide process range. On the other hand, some process models tend to oversimplify the representation of the process so that important capability checks and parameters are not considered. Aggregate process plans are generated to the detail of multiple machining steps to produce component features, with a particular machine tool selected for each process step. Thus, in turning, for example, the process should be modelled at the level of the individual passes of the tool, since the depth of cut of a pass relates to machinability constraints and to quality achieved. Where the concepts of roughing and finishing processes are to be used, the model should support the separation of these so allow realistic process plan sequences to be produced.

To achieve this level of detail in process modelling, CESS uses a three level model of process plan objects. These will be discussed in detail in Chapter 7, since they are concerned with the routing functionality and the route model. The three classes are: *Jobs*, *operations* and *elements*. Jobs and operations combine multiple processes together to create parts. Elements are defined by a single process (Figure 5.8), and are the instances of processes in the process plan. Several elements may be used to generate a single feature, using combinations of different processes if necessary.

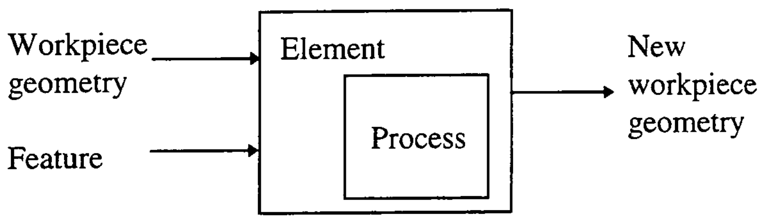


Figure 5.8: System model of an operation Element

5.3.4 Machining Process Model Structure

The process model structure consists of a single hierarchical classification. The process taxonomy classifies process types according to the similarity of the process as shown in Figure 5.1. Each process is modelled using the same class structure so that the system can treat all processes in the same way. Thus, the higher level process classes define the common attributes and methods which must be defined for each individual process. A specific detailed process class will be more complex, since there will be further attributes which are used for internal purposes (such as the process parameters, which are used as variables in the calculation of the processing time). In addition, the simple methods to calculate the costs and the constraints on the process may be linked to more complex sets of methods in the case of any given process model. However, this complexity is not seen by the program accessing the model, since the object is able to perform the calculations in its own right.

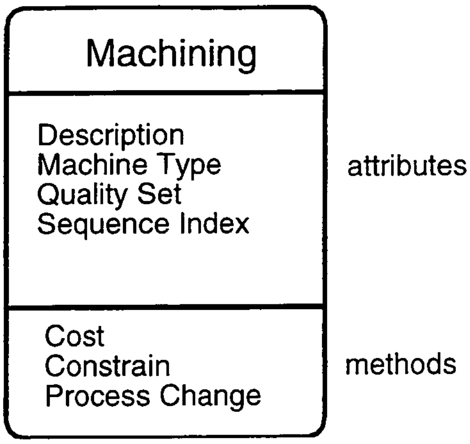


Figure 5.9: Generic process class

The generic process class, shown in Figure 5.9, does not have a large number of attributes, however the intermediate level process classes, such as drilling, turning and milling, are defined in more detail. The key attributes shared by all machine classes are the *machine type*, which identifies the class of machine tools which can perform the process; the *quality set*, which is used to define the sequence of quality level sub-processes available for the process, if any, and the *sequence index*, which is used sequence the operation elements during process planning. The methods which are present for all processes are the cost calculation method, the constraint method and the process change method which is used during planning to identify the roughing process suitable for the particular process.

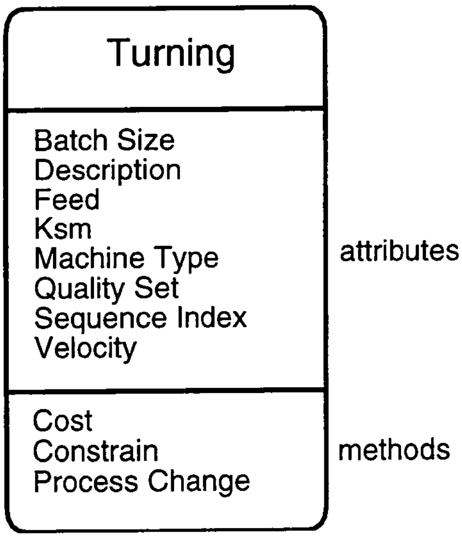


Figure 5.10: Example intermediate process class: Turning

Intermediate process classes have all the attributes which are common to each of the detailed types of that process. Thus, the turning class, shown in Figure 5.10, has attributes relating to the cutting conditions and other process parameters.

5.3.5 Process Quality Levels

Many processes are divided up into different *quality levels*, which represent the ability to modify the process parameters in order to enhance the quality of the process, usually at the expense of cost and time. The sub-processes store the limits of process quality for each level, together with information on processing alternatives for each feature. This attempt to model the use of roughing and finishing represents a simplification of the processes, since in most cases there is a continuum of parameter settings which could be selected. However, the method adopted mirrors the traditional process planning solution to this problem of adopting known levels, since to select the ideal value for each one would require too much planning time and management of the resulting data. It is, therefore, felt that the solution adopted is satisfactory for the purpose of predicting realistic processing times and costs.

The process model of CESS divides some processes into different *quality levels*. This is done where the manufacturing economics dictate that it is advisable to alter the processing conditions of the process so as to balance the accuracy and quality of the process with the cost and/or processing time required. Specialist finishing processes (such as polishing and lapping) and processes with fixed parameters (i.e. the quality level cannot be altered by changing process settings) will only have a single process quality level, and thus be represented by just a single class within this data structure. Drilling is an example of the latter process type, since there is no economic benefit to be gained from altering the cutting conditions away from the optimum for best quality of production. In most cases, however, the process parameters permit variation to improve quality at the expense of cost or time. For example, for a given tool, the surface finish in turning is directly related to the feed rate (in mm/revolution) due to the ridges produced by the cutting tool. To reduce the surface roughness and therefore increase quality, the feed rate must be reduced, thus slowing down processing time and therefore increasing cost. There are additional ways to increase turning quality, particularly through reducing depth of cut, that have a similar effect on time. When quality can be varied, therefore, it is important that the process planner considers the process at an appropriate level of quality according to the design. Processing can be split into roughing, where "low quality" process parameters are used in order to achieve low cost and near final shape geometry, and finishing, where "high quality" parameters are chosen to complete the feature.

The use of multiple process quality levels allows process plans to be both cost efficient yet of high quality. Where multiple quality levels are being applied to individual features, it follows that the creation of a single feature requires more than one processing stage. The quality type model provides the instructions to the automated process planning function on which processes should be selected to perform the roughing function for each process quality class. This structure allows the system to define multiple processing stages for a single feature. Whilst the code is written in a generic fashion to allow any number of processing steps for the manufacture of a feature, in practice it is thought that a feature would rarely exceed five processing stages. The decision about the number of steps to use is made by the system on cost grounds.

For each process, the quality level model defines the way in which process parameters can be varied to alter quality. Some processes are broken down into a number of different quality levels, representing the use of the process for roughing and finishing. Other processes have only a single level of use, typically at a finishing quality level, for example honing. The use of these quality level class divisions for each process allows the system to model more accurately the quality which will be produced from each process. Because the cost models can be modified by the parameters of the quality level classes, the system can accurately reflect the cost implications of specifying a particular quality level on a design feature. When performing a high quality finishing operation, metal cutting processes are far slower than for rough cutting where larger forces can be permitted.

For each quality level the following information, which has been determined from the tables shown in Appendix B, is defined:

Depth of cut dimension   Each process has a particular direction of material

	removal from which the depth of cut is calculated. This is designated the depth of cut dimension. For example, in longitudinal turning this is the diameter.
Maximum depth of cut	Each quality level will have a maximum depth of cut for which it should be used. Large depths of cut will require the use of a roughing process. The decision to use a roughing process is based on cost grounds.
Ideal depth of cut	Where roughing is to be performed beforehand, it is necessary to specify an amount of material to leave for the finishing process. This parameter gives the material depth which should be left if this process is to be the finishing process.
Tolerance Limit (IT)	Each process can achieve a particular quality level. This is specified as the standard tolerance interval (and surface finish). This parameter is used as a constraint, to filter out unsuitable processes during the selection process.
Minimum Tolerance (IT)	In order to be used, some processes require that the dimensions of the workpiece are within certain set boundaries. This parameter gives the required starting quality of the workpiece for the process.
Surface Finish Limit	This parameter defines the best achievable surface finish using the process under normal conditions, at the given quality level. This parameter is also used as a constraint in process selection.
Surface Finish Requirement	In certain cases, the process cannot be used unless the surface finish of the part is better than a given level, specified by this parameter.

Roughing Quality Level/ Process	If the initial workpiece conditions are such that a roughing operation is required, this parameter defines the type of process and the quality level for roughing.

The decision as to whether a process requires a roughing operation before it can be used is made using three criteria: depth of cut, initial tolerance and initial surface finish. Usually, it will only be the depth of cut which has any effect. However, if any of the limits of maximum depth of cut, minimum tolerance and surface finish requirement are not met, then roughing must be used. It is possible to have up to five quality levels specified for a single feature, and thus five separate operations carried out. These may be all of the same process type, or may use multiple process types at several levels. Table 5.1 shows an example of multiple roughing stages being used to achieve a high quality. In this example, a feature with a depth of cut of 100mm is required to be made using honing, in order to achieve a very high surface finish. Since honing requires a smooth workpiece, grinding is required. Grinding the entire depth would be uneconomic, however, so turning is used to remove the bulk of the material. Three stages of turning are used to achieve the required accuracy and speed.

Table 5.1: Example of multi-stage processing

Sequence	Process	Quality level	Depth of material removed (mm)
1	Turning	Roughing	95
2	Turning	Semi-finishing	3.5
3	Turning	Finishing	1.1
4	Grinding	Roughing	0.05
5	Honing	n/a	0.01
		Total depth:	100

5.4 Process time algorithms

This section describes the derivations of the generic process time models for turning, boring and milling. These generic models form the basis of the detailed process models for particular types of the main process, for example slot milling is a special case of milling, and taper turning is a special case of turning. The variations of each particular model from the generic model are set out in Appendix C, together with full descriptions of the process hierarchy. The CESS process time algorithms are based on aggregate process models, which operate when the full process planning details are not available, and yet a process time estimation must be calculated. Naturally, this estimate should be as accurate as possible. To achieve this, the models use calculations based on the processing geometry where possible, using automatically selected *standard cutting data*. The standard cutting data are values for the process parameters such as feed rate and depth of cut. This data is based on averages of tool manufacturers' data ([Flores 1994], [SECO, 1996], [Sandvik, 1996]). Cutting data selection requires a process planning strategy, however and in order to promote uniform treatment of the different processes, the same strategy has been applied in each of the process algorithms. Process parameters are set such that the maximum power of the machine tool is used, subject to the relevant constraints. This should result in the most efficient use of each machine tool and short cutting times. In effect, therefore, the cutting parameters are optimised, starting from a minimum set of fixed cutting parameters which relate to the workpiece material.

5.4.1 Turning and boring process time algorithms

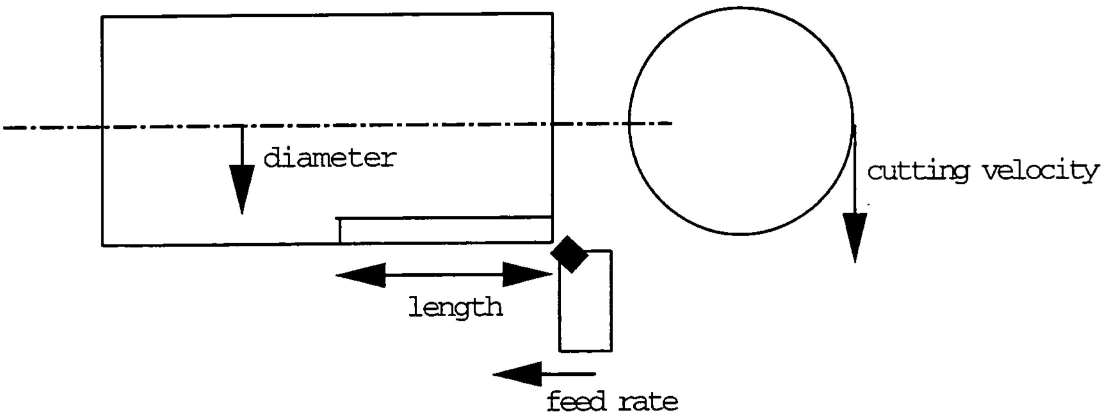


Figure 5.11: Cylindrical Turning

In order to generate an optimised process plan for a turning or boring operation considerable computation is required, consisting of many iterations through the possible cutting conditions until the best value is selected. To attempt this during the assessment of aggregate production plans would be infeasible, since the system is required to assess a very large number of potential process plans. Detailed process optimisation is not possible, since all data on the design is not present and if the data were available it would take too long. In order to provide a fast response to the queries of the designer from an incomplete model of the product, aggregate process plans have been developed.

To analyse a turning operation one must know several parameters. At any instant, if the speed of rotation, the feed rate of the tool (in both radial and axial directions) and the position of the tool are known then the forces, power requirements and material removal rate of the process can be calculated. In most cases, the rates of feed are constant, particularly in the axial direction, as is the speed of rotation. The feed rate is expressed as the distance moved per revolution, and the cutting speed defines the rate of revolution. Therefore, the cutting time, t, is given by the length divided by the product of feed rate and speed of revolution, or, alternatively, as:

$$t = \frac{l.\pi.D}{1000.v.s}$$ **Equation 5.1**

where: t = time (min)

l = length (mm)

D = cutting diameter (mm),

v = cutting velocity (m/min)

s = feed rate (mm/rev)

The critical parameters are therefore the cutting speed, the feed rate and the cutting diameter. The CESS parameter selection algorithm depends upon the power consumption. The power is given by the relation:

$$P = \frac{K_{sm}.v.a.s}{60000.\eta}$$ **Equation 5.2**

where P = Power (W)

K_{sm} = specific resistance to cut (N/mm^2),

v = cutting velocity (m/min)

a = depth of cut (mm)

s = feed rate (mm/rev)

η = efficiency

K_{sm}, the cutting force per unit chip cross section, is a material property related to the hardness, which is available from machine tool manufacturers' data books. The process planning algorithm must select values for *v*, *a* and *s*, such that the maximum machine power is used. Cutting power is limited, however, by the machine constraints on r.p.m and table feed (limiting *v* and *s* respectively). Furthermore, if only one cutting pass is required, the depth of cut, *a*, will be fixed, so the power will often not approach the machine's capacity.

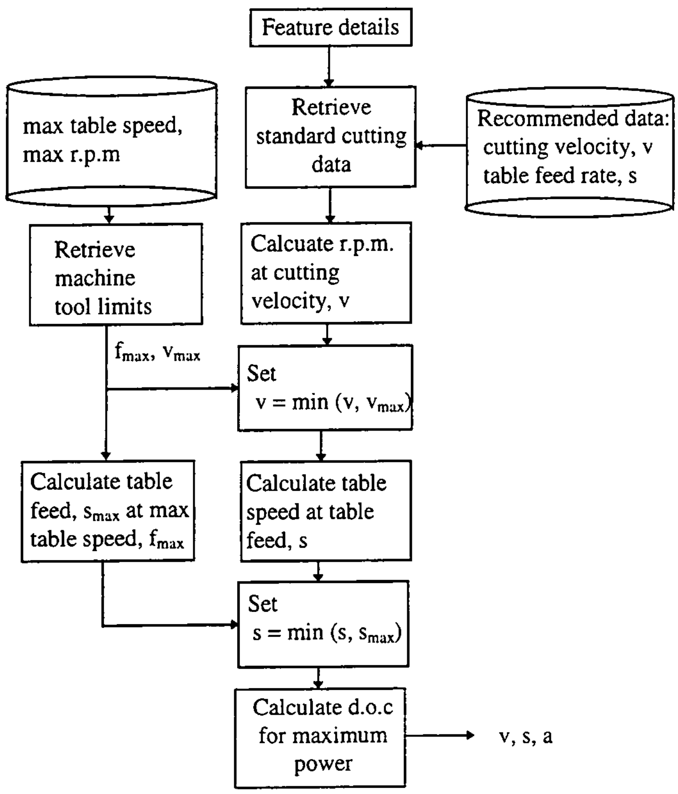


Figure 5.12: Turning parameter selection algorithm

For the CESS models, it was decided that a parameter selection algorithm which minimised the amount of calculation was necessary, since the aggregate process model may be run many times for one product. It was therefore decided to adopt a principle of using standard values of *v* and *s*, and varying the depth of cut, *a*, such that maximum power is used. For each value, however, the constraints are to be checked so that the parameters are always feasible. There is no element of iteration in the model, however, and in some less common cases this algorithm may result in sub-optimum solutions by unnecessarily limiting one parameter. However, since this merely results in an overestimate of the time, making the model more conservative, it is thought to be acceptable. The parameter selection algorithm is shown in Figure 5.12. This selection algorithm results in the swift selection of feasible cutting data which ensure that the process is operating close to the performance envelope of the machine tool, subject to the use of recognised standard data so that efficient tool use is ensured.

5.4.2 Milling

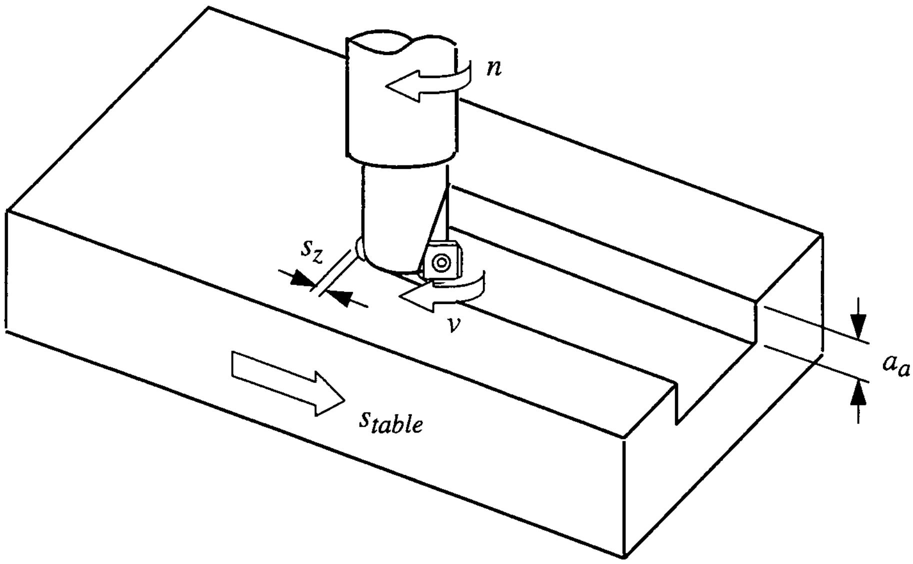


Figure 5.13: Milling

Milling is a process whereby a cutting tool is rotated at high speed, whilst being progressively fed through the workpiece in a direction perpendicular to the axis of the cutter (Figure 5.13). Cutting takes place on a number of surfaces of the tool, the teeth, as the tool rotates. Milling machines are capable of moving the cutting tool along a number of geometrical axes in order to create a particular profile, and are therefore able to generate a large number of feature geometries. In addition, a variety of tool types are available, each of which is designed for a range of operations. The milling process is therefore modelled within CESS as a set of sub-processes, based on the shape of the cutting tool. A milling tool almost always has more than one cutting tooth; tools are available in shapes ranging from cylinders to discs. Milling is a complex process to model, since each cutting tooth is in contact with the workpiece only intermittently. This means that the cutting process involves a series of impacts followed by relatively smooth cutting and then exit from the workpiece. This distinguishes the process from turning, in which the cutting is mostly continuous, resulting in widely differing cutting parameters and capabilities.

Milling is a particularly difficult process to optimise because of the large number of parameters which may be changed. Whereas in turning there will only be a single cutting edge, in milling the process planner is faced with a choice of tools, each of which may have a different diameter, with either two, four, six or even more cutting teeth, and therefore a different rotational speed to supply the same cutting velocity. Moreover, with modular carbide tools, each tool may hold a variety of cutting inserts of different geometries and grades. Similarly, the feed rate for milling must be specified *per tooth*, which means that a different table feed rate is required for each tool. In milling there are two directions of depth of cut, the radial and the axial, either of which may be used to reduce cutting rate. However, the geometry of the process means that the effects of reducing one depth of cut are different from the other. Optimisation of milling cutting conditions during tool selection therefore requires an iterative selection system which is capable of adjusting several parameters at once.

Empirical evidence and the work of several researchers ([Oberg, 1992], [Boothroyd and Knight, 1989]) suggests that it is possible to establish a theoretical metal removal rate for the various machining processes. In the case of turning, this value is of little application, since the geometry of the process means that the metal removal rate is nearly always limited by one of several machine tool and workpiece related constraints, including such factors as vibration, work-holding forces, workpiece deflection, machine tool rotational speed and table feed rate. The theoretical metal removal rate will not be achieved, or if it is, then only on a single cutting pass. Since turning is usually a multi-pass process and process plans are seldom optimised to the level of altering conditions and tools for subsequent passes, assuming that this metal removal rate is achieved would clearly under-estimate the cutting time. Similarly, in drilling, the process is limited in factors such as depth of cut by the requirement to use a tool the size of the feature and therefore it is unlikely that the limiting power will be reached. However, in the case of milling, the process is far more flexible, precisely because of the factors which make it difficult to optimise. If an optimum plan is produced, therefore, it is likely to achieve the maximum metal removal rate according to the machine power.

The CESS milling time algorithm is based on the premise that the number of available tools for a given milling job is very large: if a process planner has the ability to choose any tool in the catalogue, then it is nearly always possible to produce an optimised process plan which uses the full capacity of the machine tool. This machine tool capacity means that the process will be set to operate at one or more of the boundaries of the machine tool, such as cutting power or table feed rate. For large features, the actual tool selected should be less important than selecting the correct cutting parameters for the tool, whilst for small features, the process planner will be constrained to choose a tool which is of a size compatible with the feature, and to use only a single pass. Thus for small feature, it will not be possible to use full machine power, since the table feed will constrain the power used.

Whilst allowing the planner complete freedom to select cutting tools may lead to excessive numbers of tools being required, and thus very high tool management costs and frequent machine tool changes, there is sufficient tool capacity available in most machines to accommodate optimal tool sets, particularly since there will be considerable overlap between optimum tools for features of similar sizes. Selection of the correct tool should be given primary importance in those cases where it would affect the processing time. Except in the case of very low volume or very high variety manufacture, the process plan should be optimised with the best tools for the job. Most modern milling machines and machining centres have tool magazines capable of holding upwards of thirty tools. These machines are likely to have a comprehensive tool set which exploits the capabilities of machine, such as the power, for the range of materials for which it is used. The principle of selecting an "ideal" tool is applied consistently for all features and machine tool at aggregate level. This allows decisions to be taken and priorities to be set at that level using a consistent and systematic method. From these plans it is possible to select a tool-set for the machine for all components which visit it. If at the detailed level a different policy is operable concerning optimisation and tooling, the cutting data will need to be recalculated.

5.4.2.1 Milling optimisation tests

The contention that it is generally possible to optimise milling operations such that the theoretical maximum cutting rate is achieved has been tested using a detailed automated process planning system for milling, OPTIMUM (Carpenter, 1996). OPTIMUM is a computer based machinability assessment and tool selection system which has been developed at the University of Durham. It was particularly suitable for the task of testing boundaries of milling efficiency, since it is an experimental system in which the harshness of cutting data can be controlled. Additionally, the system has access to a comprehensive database of both cutting tool holders and inserts entered from a manufacturer's catalogue (SECO, 1996). In order to test the proposed model for milling, it was therefore decided to generate a series of process plans for individual features using the OPTIMUM system, and to analyse the parameters which were suggested, the tools selected and the cutting times. For each run of the optimisation system, the inputs were the feature type, its dimensions and the key constraints of the machine tool, namely; maximum cutting power, speed and table feed as shown in Figure 5.14.

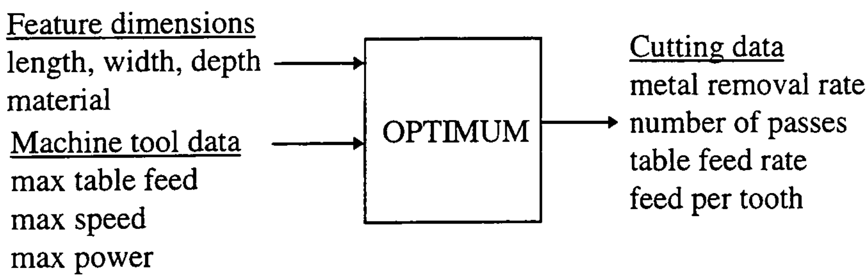


Figure 5.14: Testing procedure using OPTIMUM system

The purpose of these tests was two-fold: to identify the relative importance of the various constraints on milling parameter selection during optimisation and to test whether the material removal rate for a given power and material was constant. If the material removal rate is divided by the power, the specific material removal rate, V_p (cm^3/min.kW) can be obtained, which allows comparison of the results for the different machines. In this discussion, the specific material removal rate predicted has been plotted against the feature width for three different cases:

- Varying feature dimensions (workpiece material grade and machine tool fixed).

The feature dimensions are of critical importance to the feasibility of using maximum machine tool power in milling. When the feature is small, then it is not possible to use full power, since this would imply feeds and speeds in excess of the machine tool limits. The two key dimensions for most milling features are the depth and the width, which correspond to the axial and radial cutting directions, respectively. The length does not affect parameter selection greatly. In Figure 5.15 it can be seen that the width of the feature has a significant effect on the material removal rate (expressed as V_p) which can be achieved for a given feature length and depth. A characteristic curve is generated for material removal rate and when the width is very low, then V_p is proportional to the width. It is clear, however, that a limiting value of material removal rate is reached, which is due to the power limit of the machine tool. The graph has a saw-tooth appearance, consisting of a series of rising lines of decreasing gradient, which correspond to the different tools within the database. If the number of tools were greater, then the saw-tooth effect would tend to be reduced and the material removal rate would remain nearer to the maximum value. The OPTIMUM system, being an experimental system did not have the full set of tools available at time of the testing and so the curve is not as flat as was expected.

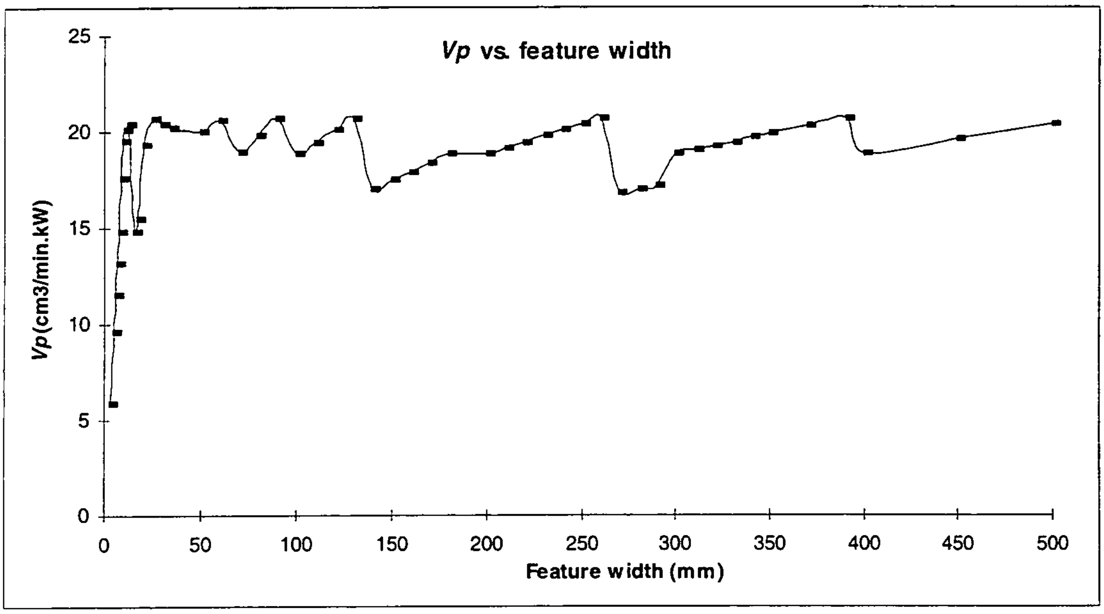


Figure 5.15: Variation in specific material removal rate with feature width

If the feature depth is considered, however (Figure 5.16), it can be seen that the effect on the material removal rate is less critical. Whilst the feature depth is still a factor in limiting the material removal rate for low width features, it rapidly becomes irrelevant as the feature width is increased. The figure shows that for the deeper features the gradient of the initial curve is steeper than for the more shallow features, but they both reach the same maximum value after about 10mm, which must be considered small for general machining. The same "saw-tooth" pattern that was seen in Figure 5.15 is found in this figure, this is again due to the lack of tool variety in the database for small tools.

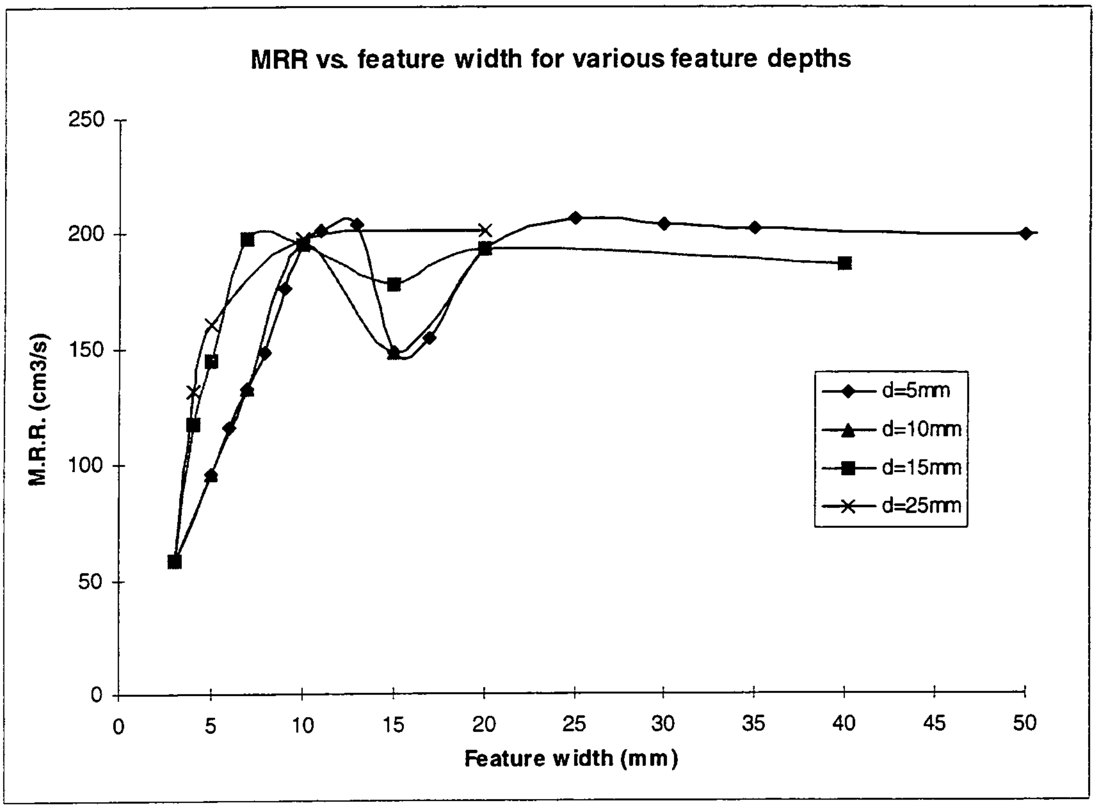


Figure 5.16: Effect of feature depth of milling efficiency

- Varying workpiece material grade (feature depth and machine tool fixed).

When the workpiece material is varied, the material removal rate for a given power changes: each material grade has its own specific material removal rate which is the removal rate per unit of power assuming that the process is not constrained by geometry or machine characteristics and that the cutting conditions are suitable. If the material removal rates for various feature widths are plotted for different materials (Figure 5.17), it can be seen that each material specific curve has the same shape, but the magnitude of the removal rate is different.

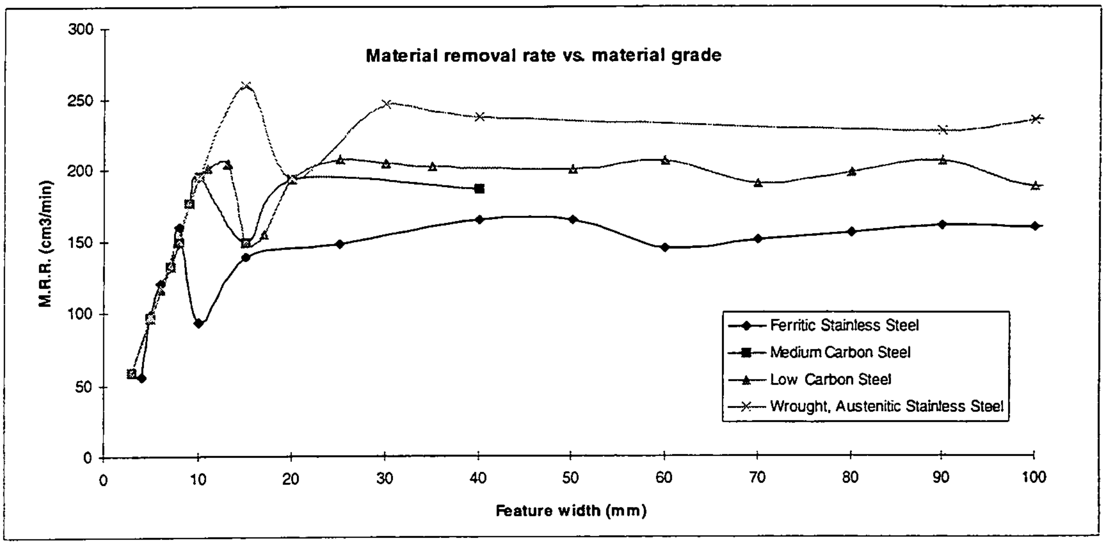


Figure 5.17: Effect of material grade on milling rate

- Varying machine tool (workpiece material grade and feature depth fixed).

The key machine tool factor was soon found to be the table feed: when a low table feed limit was present, the material removal rate was constrained far more frequently. Three machine tools were used in the tests:

Table 5.2: Machine tools used in testing

Machine tool	Power (kW)	Table feed (m/min)
A	16	15
B	10	4
C	5.5	12

It can be seen below (Figure 5.18) that V_p reaches a plateau region for features wider than a certain value, depending on the machine tool. For the machine tools with high maximum table feeds (A and C) this, transition is lower than when the table feed is limited. The table feed controls the transition between the two areas of the graph: below a certain width, the parameters are selected at the maximum table feed, whilst above that width, the table feed is not at maximum, but the machine power is.

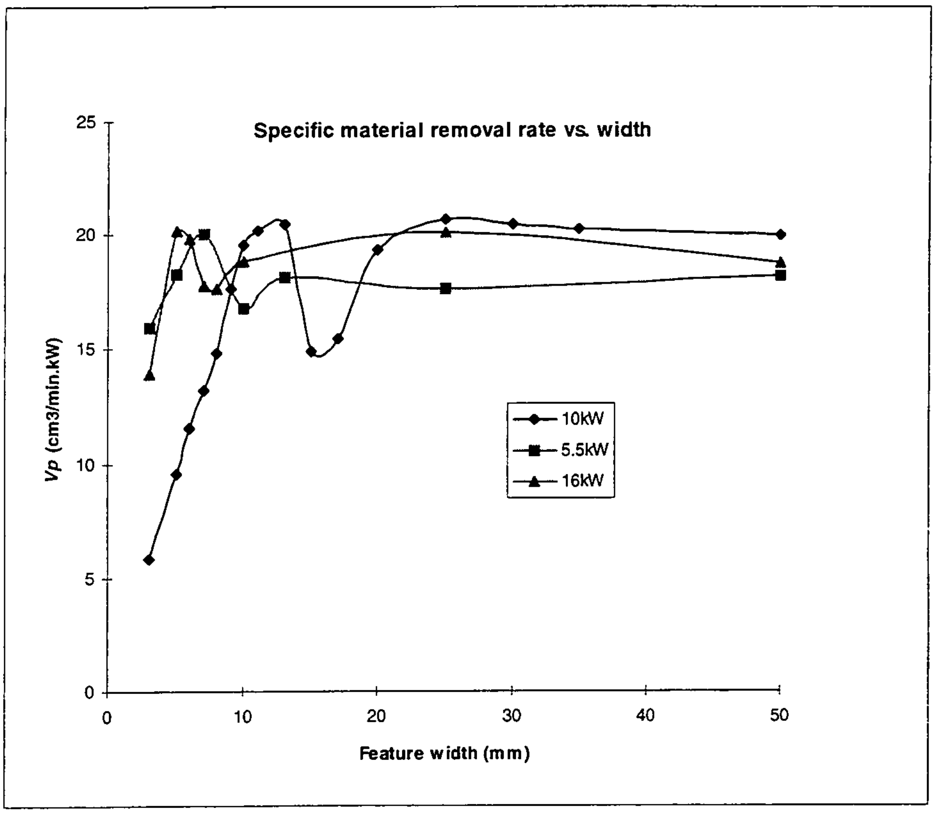


Figure 5.18: Effect of variation of machine tool characteristics on milling rate

Whilst the above examples are all based on face milling, the results for other processes were broadly similar. The effect of feature width on the material removal rate is less simple with enclosed milling operations such as slotting, since the tool selection is limited by the confines of the feature. Slot milling also tends to be a more complex process to plan for, since the conditions will vary between the initial cut, which is a full immersion cut, and subsequent cuts of the same axial pass. The OPTIMUM system is designed to consider the selection of different cutting conditions and tools on the initial and subsequent cutting passes. It is therefore more difficult to represent the results in the form above. The studies carried out for slot milling where based on calculating the average material removal rate for the entire feature.

5.4.2.2 Milling process algorithm

The algorithm for the calculation of milling processing time is shown in Figure 5.21. The process time is calculated from the processing rate according to the simple relation:

$$t = \frac{d.w.l}{M}$$ **Equation 5.3**

where d = feature depth (mm),

w = feature width (mm),

l = feature length (mm),

M = material removal rate (cm^3/min).

The material removal rate, M, is constrained by two factors: feature geometry and cutting power. The cutting power constraint is calculated from the specific material removal rate which is stored for each material grade (Appendix D):

$$M_{pow} = \eta . P . V_p \qquad \textbf{Equation 5.4}$$

where V_p = Specific material removal rate for milling (cm^3/kWmin).

The geometry constraint is dependent on the maximum table feed rate of the machine and the relationship between the feature size and the tools available for the process. Each milling process has two parameters, cutting depth and diameter, which determine the maximum size of the tools available. To calculate the maximum removal rate according to the geometry it is necessary to determine the greatest cutting area which is feasible, and to multiply this by the maximum table feed rate:

$$M_{geom} = a_{a\max} . a_{r\max} . s_{\max} \qquad \textbf{Equation 5.5}$$

where a_{amax} = Maximum axial depth of cut (mm),

a_{rmax} = Maximum radial depth of cut (mm),

s_{max} = Maximum table feed rate for machine (m/min).

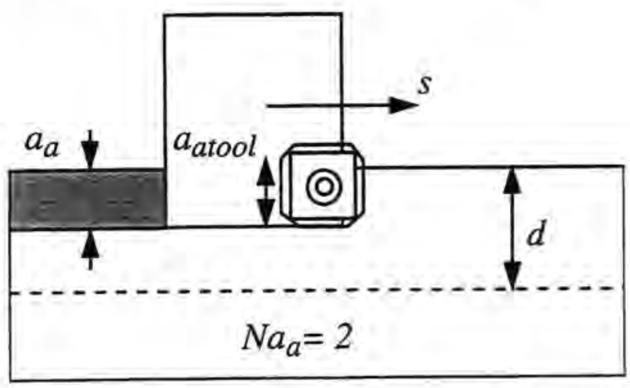


Figure 5.19: Axial depth of cut in milling

The cutting area is the product of the maximum depths of cut in the axial and radial directions. The axial depth of cut, shown in Figure 5.19, is calculated using the same method for both open and closed features:

$$n_a = \frac{d}{MIN(d, a_{atool})}$$
$$a_{a\max} = \frac{d}{n_a}$$
Equation 5.6

where n_a = number of axial passes,

a_{atool} = maximum axial depth of cut for tool (mm).

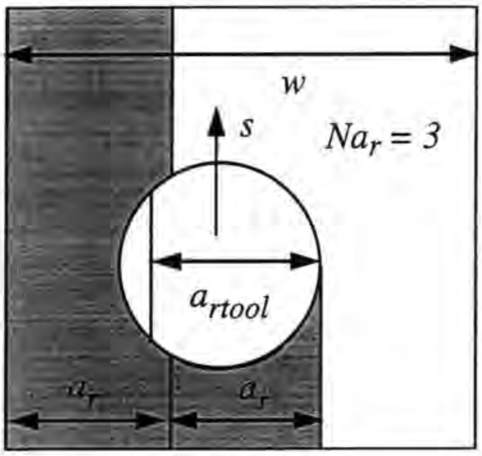


Figure 5.20: Radial depth of cut in milling

The method for calculating radial depth of cut, shown in Figure 5.20, will vary depending on whether the feature is enclosed or open, since for enclosed features, the radial depth of cut must be less than the feature width, whereas in open features it can be greater. Also, the cutting depth must divide the total feature depth into an integer number of passes. For open milling, such as shoulder milling, the radial depth of cut is given by:

$$n_r = \frac{w}{MIN(w, a_{rtool})}$$
$$a_{r\max} = \frac{w}{n_r}$$
Equation 5.7

where n_r = number of radial passes,

a_{rtool} = maximum radial depth of cut for tool (mm).

For enclosed features, the radial depth of cut is restricted by the feature walls. Since in general a tool of exactly the same width as the feature will not be available, there must always be at least two radial passes. This is reflected in the equation for depth of cut by taking half the feature width as the maximum cutting width:

$$n_r = \frac{w}{MIN(w/2, a_{rtool})}$$
$$a_{r\max} = \frac{w}{n_r}$$

Equation 5.8

This algorithm calculates an estimate of processing time assuming that the process conditions will be optimised, without the need to perform a lengthy optimisation procedure for each machine option. It is therefore very suitable for use in an aggregate process planning situation. The process times generated can serve as a target or benchmark for process planners and give a quick cost indication for design purposes.

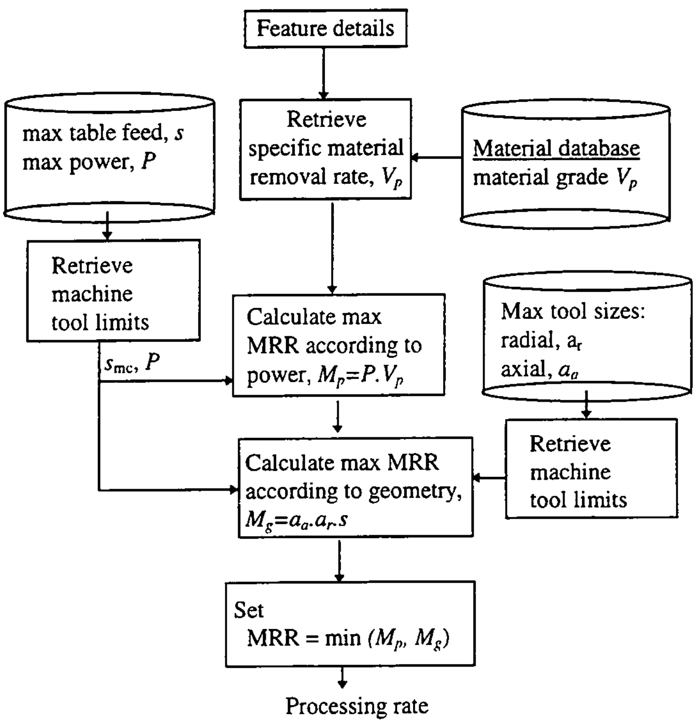


Figure 5.21: Milling processing time algorithm

Each milling process has a slightly adjusted version of the generic milling time algorithm. The method can be readily adapted to the various sub-processes of milling such as chamfering, through the modification of the geometry constraint. The use of tool size limits is particularly useful in representing the differences between process types, with specialist tools only available in the case of specialist processes. As an example, facing tools are available in diameters up to 500mm, but have a maximum cutting depth of only 15mm. Edge milling cutters, in contrast, are available with diameters up to only 100mm but have cutting depths of up to 77mm. These limits are shown, along with details of the changes to equations, in Appendix C.

5.4.3 Grinding

Grinding processes consist of moving an abrasive surface against the workpiece to remove the material in a gradual process. A detailed study of grinding has not been undertaken during this work due to the limited applications of grinding in high volume production: it is primarily used as a finishing process. A basic model of processing time has been adopted, however, which is similar in approach, though less complex, to the turning process time model. A similar approach is applied to all the sub-classes of grinding, with minor variations between cylindrical and surface grinding. The example shown here will be for cylindrical traverse grinding.

The time required to grind a cylindrical feature is given by the relation:

$$t = p\frac{l.\pi.D.1000}{w'.v_w}$$ **Equation 5.9**

and

$$w' = \frac{w}{\alpha}$$ **Equation 5.10**

and

$$p = INT(\frac{d}{i}+1)$$ **Equation 5.11**

where p = number of passes,

l = feature length (mm),

d = feature depth (mm).

w = grinding wheel width (mm),

w'= infeed (mm/rev)

v_w= work speed (m/min)

i = depth of cut (mm),

α = quality factor,

Standard work speeds are defined for each material within the CESS database, a typical value being about 30m/min for plain carbon steels. The quality factor, α, is a constant for a particular grinding quality level, with values ranging from α=2 for rough grinding to α=6 for finishing. The depth of cut values are again defined for the process class. Typically depth of cut will be between 0.013mm and 0.05mm. The main difficulty in determining suitable grinding parameters to select for this model lies with the wheel selection itself. For detail planning of grinding operations, the wheel selection must be carefully made according to many factors. There are a large variety of wheels available with various dimensions. However, for aggregate process planning, it is only necessary to select certain values relating to the wheel. In this case, the system assumes that a grinding wheel with a width halfway between the minimum and maximum available will be used, unless this is precluded by the size of the feature, in which case the smallest wheel width is used instead.

5.4.4 Drilling

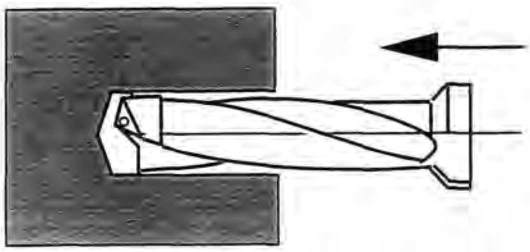


Figure 5.22: Drilling

The drilling process (Figure 5.22) is similar to milling, using a rotating multipoint cutting tool. The direction of feed is limited, however, such that the tool is only moved along its axis (whilst in contact with the workpiece). The drilling process consists of moving an axi-symmetric cutting tool along its axis of symmetry into the workpiece, to form a cylindrical hole. The cutting tool has a number of cutting edges on the bottom and sides, and a means of removing the workpiece material as chips. In general drilling machines operate vertically, although in CNC machining centres or manually, drilling may be performed at any orientation and when performed on a lathe the orientation is horizontal. There are several sub-processes within the class of drilling, each of which produces a slightly different feature set using a modified tool. All drilling sub-processes use the same generic drilling time algorithm, with the differences between the types being implemented through the use of alternative cutting conditions, and through the method which calculates the depth of cut of the operation.

The cutting time for drilling is given by the drilling length divided by the axial feed rate, which can be expressed as:

$$t = \frac{(l+d)}{N.f_n}$$ **Equation 5.12**

where N = spindle speed, (rev/min),

f_n = down feed (mm/rev)

Since in general the cutting tool (drill bit) has a point angle, the total drilling length will be greater than the depth of the hole. Typically the actual drilling length is increased by the diameter of the hole, as shown. To calculate this time, the cutting parameters must be selected. In accordance with the parameter selection strategy previously outlined, this task is accomplished by combining suggested average values from the material database with an optimisation based on the machine tool power, as shown in Figure 5.23. Power use in drilling is given by the relation (Sandvik, 1996):

$$f_{pow} = \frac{P.\eta.60}{a_p.K_{cfz}.v_c}\left(\frac{1}{1-{a_p}/{d}}\right)$$ **Equation 5.13**

where f_{pow} = downfeed at maximum power (mm),

a_p = effective depth of cut (mm),

K_{cfz} = drilling resistance (N/mm^2),

v_c = cutting velocity (m/min)

The algorithm is driven by an ideal cutting velocity, v_c for the material, which will be selected as long as this is not constrained by the maximum spindle speed of the machine tool, N_{max}. To test this, the maximum velocity which the machine tool can achieve is calculated according to the relation:

$$f_N = \frac{N_{\max}.\pi.d}{1000} \qquad \textbf{Equation 5.14}$$

where f_N = downfeed at maximum spindle speed (mm/rev),

The lower of the two velocities is assumed and the power equation is then solved for the downfeed, f_{pow}, using the calculated velocity, the machine power and feature depth of cut. The material parameter K_{cfz} relates the force required to cut to the cutting conditions. The calculated feed rate is then compared with both the maximum specified for the material and the machine tool and the lowest of these values is used to calculate the time.

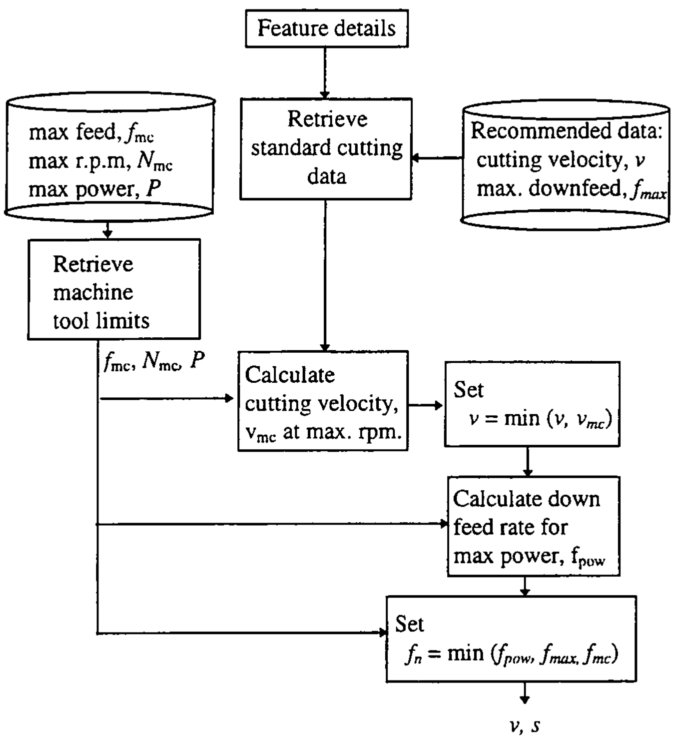


Figure 5.23: Drilling parameter selection algorithm

5.5 Aggregate assembly process modelling

This project does not cover the assembly process since CESS deals with individual component plans rather than assembly plans. However, the product model which has been developed includes the ability to model assemblies as well as components and it was therefore necessary to address the issue of generating aggregate plans for assemblies. This function would provide benefits of giving consideration to capacity factors and conflicts in resource requirements between components in the same product. In addition to combining the machining plans for individual components, a manufacturing plan for an assembly, called a *routing*, would require plans for the assembly operations. It is recognised that in many products assembly operations contribute a high proportion of the manufacturing cost. At this stage of the project, a rudimentary aggregate assembly model has been adopted. The aggregate assembly model is based on a hierarchical taxonomy of assembly methods, similar in nature to the machining process model. It is envisaged that individual process models would be created for each assembly method, such as manual assembly, robotic assembly, gluing and riveting. However, the current system does not have these models in place. Instead, the times to complete each assembly connection must be entered directly by the user. The process models exist to allow the identification of equipment requirements, and thus routing constraints. Further research on aggregate assembly models to complement CESS has been undertaken by Betteridge (1996).

5.6 Process quality modelling

This section discusses the CESS quality modelling algorithm. In order to provide realistic manufacturability analysis, it is important to provide an accurate assessment of the implications of the product design on the quality during production. Quality issues must be central to any manufacturing concern, since a product which meets all other criteria but lacks quality will ultimately be unsuccessful. It was possible in the past to accept a low standard of product quality if goods where produced and sold cheaply. Now, however, with the advent of high technology processes and the more open world market, it is possible to set up factories to produce high quality goods very cheaply in areas with low overhead costs. It is competition from the emerging economies which forces contemporary manufacturing to accept nothing less than the highest quality. In order to discuss quality, it is firstly necessary to give a definition of the term. In a manufacturing context there are two different aspects which can be covered by the word quality. These may be termed functional quality and production quality. Each of these sections is of importance and must be considered for any new product.

5.6.1 Functional Quality

The functional quality of a product is the degree to which it meets the performance criteria set out in the specification. This functional quality should be independent of the manufacturing process to be employed for the product. Examples of performance criteria would be the power produced by a motor or the strength of a bracket. In order to assess functional quality, two methods are generally used: modelling and prototypes. In a simple situation, the designer will use mathematical models of the product to calculate the required size of components. In more complex situations computer models are used, such as finite element models in aerodynamic design and simulation for dynamic mechanisms. Prototypes are examples of the product which are manufactured, usually in specialised workshops, to test aspects of the design.

The assessment of the functional quality of a design requires a specialist tool which is suited to the particular product type. This will generally embody detailed knowledge which is specific to the products and company involved. Many tools are available to perform analysis of the functional quality, including finite element packages for mechanics, dynamics, electromagnetics, thermodynamics and fluid mechanics. Simulation tools are used extensively in electronic and mechanical design. It would therefore be inappropriate for a system such as CESS to attempt to calculate the functional quality. Instead, an interface with specialist systems through the exchange of product model data could provide performance assessment. CESS would be used as a tool for analysis of manufacturability.

5.6.2 Production Quality

The production quality may be defined as the degree to which the manufacturing process produces parts which meet the design requirements. The production method will be set up to produce parts according to the dimensions given by the designer. Any manufacturing process, however, will inevitably be subject to some variation. Similarly, it is never possible to achieve exactly the size which the designer might specify: indeed, it is usually not desirable to try to produce parts to a very accurate dimension. This variation in manufacturing processes is recognised in the design system of tolerancing of dimensions. Parts are specified with dimensions and a range within which this dimension must fall on all the parts. If the dimension is outside the tolerance range, the part is considered sub-standard and is either scrapped, or sent for re-work, depending on the economics.

5.6.3 Process Capability Indices

The process capability is a measure of the degree to which the process is capable performing an operation to the required accuracy. This is a ratio of the tolerance of the dimension to the amount of deviation from the nominal value of that dimension. The most typical example of a process capability index is C_{pk}, which is defined as:

$$C_{pk} = \frac{\Delta T}{6\sigma} \qquad \textbf{Equation 5.15}$$

where ΔT = Difference between upper and lower tolerances (mm)

σ = Standard deviation of process (mm)

This is a familiar indicator in industry, and a very useful check on the degree to which the process is under control. From Gaussian law it can be calculated that six times the standard deviation corresponds to 99.73% of the samples, so a C_{pk} =1 indicates that three parts in every thousand would exceed the tolerance limits. A value below one indicates a process which is out of control, whilst a capability above one indicates a process under control. In some industries, alternative indices are more common, typically using a larger denominator such as 10σ, for instance in the automotive industry, where accuracy is required to be higher.

5.6.4 Production quality prediction

Production quality in machining is a complex property which depends on many factors: workpiece set-up accuracy, tool and workpiece deflections and deformation under machining forces, material consistency, ambient conditions, manufacturing defects in tools, machine tool control systems, human error and reliability. To model each of these factors individually in order to arrive at a deterministic model for workpiece quality would be an impossible task. It is therefore necessary to seek an alternative approach. It can be seen that many of the factors affecting workpiece quality are stochastic variables: for each workpiece in a production run, there will be small random variations in most of the factors. It is usually when several factors combine to cause the same effect that large changes in workpiece quality result.

It is necessary to decide upon an indicator to represent the quality of production. This indicator must be a useful measure of the relative quality between alternative product designs or production methods. Typically production quality is measured as the rate of work which is scrapped due to failure to meet the design specifications, or as process capability, which is a ratio representing the likelihood of a part being produced within the tolerance. Taguchi introduced the concept of quality loss as an indicator of product quality. This measure recognises the inherent cost to the company of producing parts to low standards. This cost is related both to the cost of scrapped parts and to the perceived quality of the product in the market place. It therefore aims to include both functional and production quality and requires more information than is available at aggregate level. Since it is desirable to relate the change in quality with the change in cost between design or production alternatives, however, it is appropriate to try to represent quality in terms of cost. The cost of low production quality may be calculated by determining the additional production cost of re-working sub-standard products and of producing additional products to replace those which are scrapped. This method has been adopted for the quality indicator within CESS, allowing the cost of quality to be added to the cost of production so that quality may be built into the process and machine selection algorithms directly.

To determine the quality cost, the production quality of each manufacturing operation is modelled as a rate of scrap per workpiece. Thus the maximum value of scrap rate is unity, indicating that all parts are scrapped (Typically scrap rates are less than 0.001, which approximately equivalent to $C_{pk}=1.3$). In addition to scrap rate, the system calculates the process capability index for each operation, C_{pk}, for output to the user.

The quality algorithm of CESS uses a statistical prediction method based on the assumption that production quality is a stochastic property which is the product of a set of stochastic variables. A random distribution is applied to the values of each dimension of the feature created by an operation, based on data drawn from historical examples of similar operations.

It is assumed that the distribution of dimensional values is according to the normal law, since according to the Central Limit Theorem, which can be stated as:

> "If X_1, X_2, ..., X_n are independent random variables with arbitrary distribution laws then the distribution of the sum $Y=\Sigma_i X_i$ tends to the normal law as n increases".

It has already been observed that the quality is affected by many random variables, therefore it can be expected that production quality will obey the normal law. Once the distribution of dimensions is modelled using a normal distribution, it becomes possible to predict the proportion which will fall outside the boundaries set by the tolerances. Gauss' law states that, for a variable which has a normal distribution, the probability of a sample selected at random being between a and b is given by:

$$P(a \le x \le b) = \int_a^b f(x)\,\mathrm{d}\,x \qquad \textbf{Equation 5.16}$$

where:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-m)^2}{2\sigma^2} \right) \qquad \textbf{Equation 5.17}$$

where m = mean data value,

σ = standard deviation.

To solve this function it is necessary to use a numerical integration technique. In CESS the integration is performed using the Trapezium rule with a fixed number of steps.

5.6.5 Quality algorithm

The quality algorithm in CESS uses the statistical method described above in order to calculate a predicted scrap rate (and hence process capability) for each operation element. This allows the system to determine the cost of scrap for each element, which is added to the total production cost.

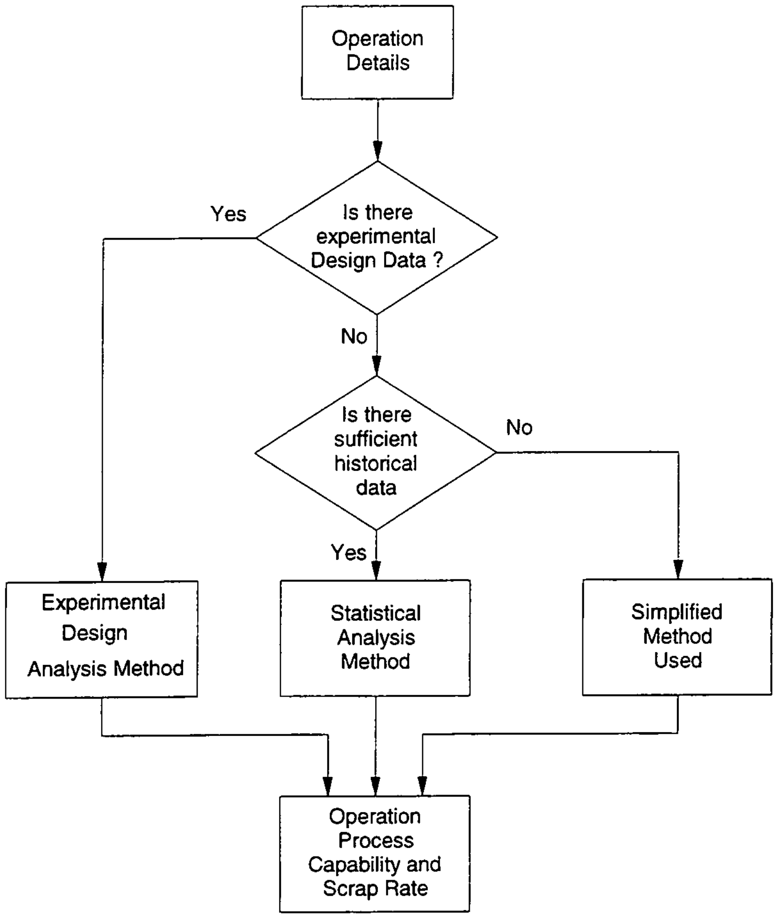


Figure 5.24: Three stages of quality

A three stage quality prediction algorithm is proposed for aggregate process planning, based on the need to make use of whatever quality information is available at any given product development stage. It has been remarked elsewhere that the amount of quality information available increases throughout the design process. In addition, where unusual process and machine combinations are selected, there may be insufficient data to generate valid distributions for the feature dimensions. Alternatively, additional information may be available in the shape of experimental design data. In this case, a more detailed analysis of quality could be provided. The proposed architecture of the

CESS quality prediction algorithm is shown Figure 5.24. At this stage only the statistical and the simplified quality prediction modules have been implemented in CESS.

5.6.5.1 Statistical quality calculation algorithm

This method uses the principles detailed in the previous section. The structure of the algorithm is described in Figure 5.25. Quality is calculated by the system at the operation element level, which represents a single processing step in the creation of a part. This element combines information from the product, the process and the resource model through inheritance of attributes and methods.

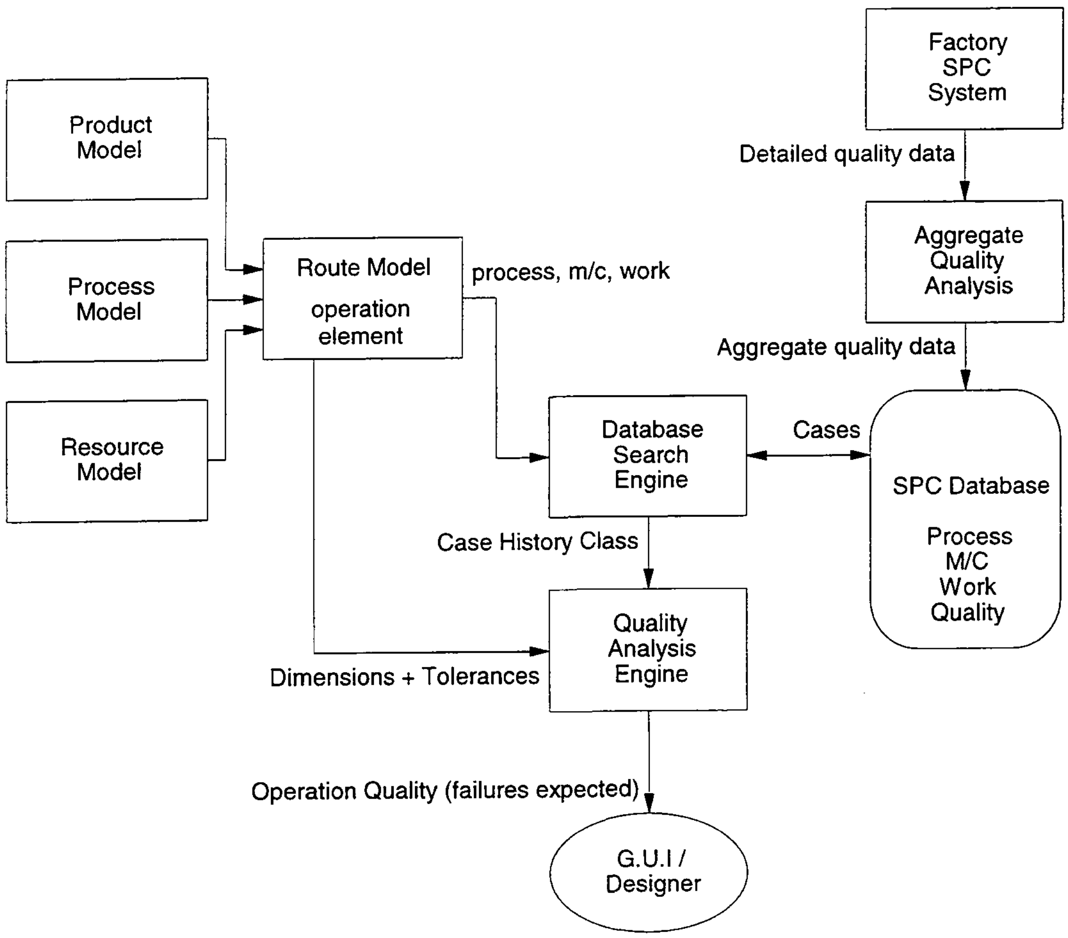


Figure 5.25: Statistical quality analysis schematic

The operation element forms the input to the quality algorithm: the element parameters are passed to the database search engine, which interrogates the quality database. This database consists of aggregate quality data which has been gathered from the shop floor SPC quality systems, and then pre-processed to convert it into a suitable format. The data is translated into aggregate form which is suitable for comparison with the aggregate information within CESS. The search engine generates a list of previous cases which are similar to the current operation element, in terms of process, feature and machine tool. This list is refined into a manageable size by extracting only the most similar and/or reliable data and then passed onto the quality analysis engine. Similarity is determined by the number of matching parameters: e.g. operations are considered similar if they use machine tools of the same class, but if the same specific machine tool is used in each case then this would have a higher similarity rating and be used in preference.

The quality analysis engine operates by calculating the average standard deviation of the historical examples. This data is then used to find the probability of the dimensions of the new operation element falling outside the allowed tolerance bands by solving Gauss's law. This is equivalent to the scrap rate of the process.

The third system module shown in Figure 5.25 is the Aggregate Quality Analysis module. The purpose of this module is to provide off-line analysis of SPC data and to extract and format the relevant data for storage in the CESS SPC database. This is a management function of the system, which falls outside the scope the project. The output of the system is defined, however, by the database structure (Table 5.3):

Table 5.3: SPC database format

ID	Feature	Dimension	Process	Machine tool	Machine class	σ	Sample size
0101	pho	diameter	drilling	Mazak11	CNC_mill	.05	5
0102	ecy	diameter	turning	Traub62	CNC_lathe	.03	10
0103	ecy	length	turning	Traub62	CNC_lathe	.14	10

Since the aggregate process planning system is designed to operate on aggregate design information, there will be features for which the tolerances have not yet been specified. In this case, the system must select a standard tolerance interval (IT) and apply it to the feature so that the process selection methods can operate at an appropriate level and a process of sufficiently high quality can be chosen.

5.6.5.2 Simplified quality calculation algorithm

In order that the quality calculation algorithm does not break down when unusual processes and machine tools are selected it is necessary to define a method to predict quality when no surrogate data can be generated. This might occur if a process is used very infrequently, or if the process is new to the company, for example. This method uses data available from previous manufacturing research to calculate estimated process capability figures. Many manufacturing texts provide general tolerance levels for process selection.

These sources (Appendix B) suggest the upper and lower tolerance limits for which a process is suitable, in terms of standard intervals of tolerance. Thus, given a capability of the process at these tolerance values, it would be possible to interpolate between them to estimate the capability of any intermediate tolerance (Figure 5.26). Furthermore, the line could be extrapolated out to estimate values outside the range if necessary. This assumes that the process capability varies linearly with the tolerance interval number. Whilst it would be difficult to prove this assertion, it should be noted that for fixed process parameters and an automated process, the distribution of dimensions produced by the process does not vary with the target tolerance. Therefore, the variation of process capability with absolute dimensional variation will be linear. In CESS the process parameters are generally assumed to be roughly fixed within a quality level, therefore it is appropriate to make this assumption for estimation purposes.

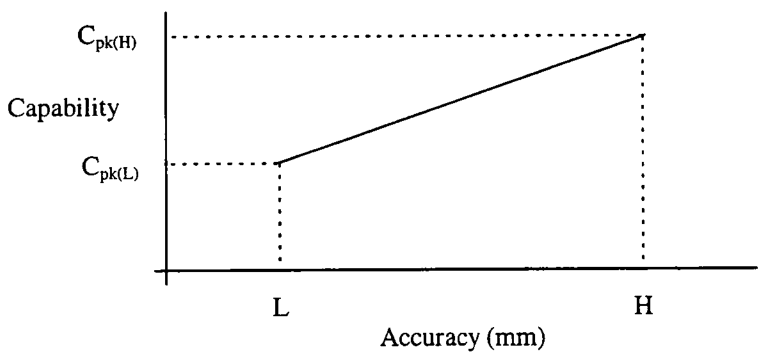


Figure 5.26: Process Capability Interpolation

The question then becomes what values of process capability should be adopted for the minimum and maximum accuracy which are available. Ideally, the process capability should be determined experimentally for each process. This would, of course, require experimentation on a range of machine tools in order to arrive at a generic figure. Alternatively, it may be noted that a process capability $C_{pk} = 1$ corresponds to a process which is considered just acceptable. In other words, this is the value at the limit of a process's suitability. This could then be adopted for the upper quality boundary. The capability corresponding to the lower value could be expected to vary considerably across differing processes. However, a single value might be adopted for the purposes of estimation.

Thus, if each process is assigned a minimum and maximum accuracy for which it is suitable, and the process capability for each of these accuracies is assigned a value, then the process capability of any intermediate value is given by:

$$C_{pk} = mT + c \qquad \textbf{Equation 5.18}$$

where:

$$m = \frac{C_{pk(H)} - C_{pk(L)}}{H - L} \qquad \textbf{Equation 5.19}$$

and

$$c = C_{pk(H)} - mH \qquad \textbf{Equation 5.20}$$

where $H = \Delta T$ at minimum economic quality (mm),

$L = \Delta T$ at maximum economic quality (mm),

and ΔT = tolerance interval (mm).

The values of H and L can be determined for a given nominal dimension from the standard interval of tolerance (IT) values. If a value is encountered which falls outside the process range, the system would have two options. The first is to reject the process as unsuitable, which is to use the quality as a constraint. The second is to use the same system to extrapolate the capability from the available range. Clearly the accuracy of this will be less valid as the tolerance value in question deviates more from the range. However, an overly accurate specification still results in a capability of less than one, so the prediction errs on the side of caution.

5.6.5.3 Geometrical tolerances

A mention should be made at this stage of geometrical tolerances, since the examples used in this thesis have mainly been size, or linear tolerances, such as length and diameter. It should be noted, however, that the same methodology which is applied for the size tolerances will also work effectively for the geometrical tolerances. Each of these also has a distribution of values which can be characterised by the standard deviation. There is one qualification, however, and that is that the definitions of most geometrical tolerances are such that there is only a single boundary value. The tolerances are expressed in terms of how much they may deviate from a particular direction or point. For example, a concentricity tolerance is defined by an axis and a distance by which the axis of the second circle may be from that axis. However, these situations can be considered as single sided versions of the simple size tolerance. Thus, the methodology for these cases can be easily adapted from that for dimensional tolerances.

5.6.5.4 Quality cost

Once the quality of individual operation elements has been calculated, the aggregate process planning can use these values to calculate the cost of the quality. This procedure is described along with the main cost calculations in the next chapter. The basic principle which is employed is to assume that for every scrapped part, a replacement must be made. Therefore the production cost is multiplied by the scrap rate plus one. However, it must be remembered that when a part is scrapped, all the previous operations are wasted. Therefore the cost which is multiplied must be the cumulative cost of all previous operation elements in the sequence. This factor means that it is impossible to calculate a valid quality cost without first determining the sequence of production. This influences the design of the process planning function.

5.6.6 Summary

The process quality prediction method uses a statistical method based on proven factory data and an understanding of the multiple influences on quality to return useful information about the expected production quality of the product design. It is important to provide such quality assessment to aid in guiding the specification of design tolerances, as well as the selection of capable processes and resources.

5.7 Conclusions

A set of process models have been developed which allow the system to automatically assess the manufacturability of a given design from the product model. The process models provide information on the processing time required to produce features depending on the feature dimensions. In addition, the setting up times required and the material and energy costs of the processes are considered to allow the costing of the use of each process alternative.

Chapter 6

The Resource Model

6.1 Introduction

This chapter details the resource model of CESS. The resource model represents the tools, equipment and facilities available to the company to produce the product. A resource model is a prerequisite for aggregate process planning, which considers both the processes and the equipment which can be used for manufacture.

Resource information which is required for aggregate process planning includes both equipment and organisational data. The CESS resource model uses a hierarchical resource model based on the concept of factories; a factory is a production unit which consists of a number of manufacturing cells. Within the factory, information on transportation, storage, processing equipment and labour resources are modelled. The CESS model supports the use of multiple factories, which can represent either alternative locations for manufacturing the product (useful for make or buy decisions), or alternative configurations of the same location (useful for facility design).

The resources model allows the user of CESS to customise the system to suit their own requirements. A generic modelling scheme has been developed which can be applied to any factory system. The model structures are populated with data about the resources present in the particular factory.

6.2 Resource elements

A number of elements are necessary to constitute a manufacturing facility. Many of these must be considered in order to develop process plans which use that facility. The following list discusses each of the resource elements and the possible methods for modelling them within a computer system:

- Factory

A factory is a facility for the production of products. Factories are usually located at a single site under the administration of one production manager. Typically a factory may be on the same site as the product development team, although frequently products are manufactured elsewhere, particularly when the same product is made at several locations, each close to a particular market. For the purposes of aggregate process planning, it is assumed that only a single factory is to be used for the manufacture of a product: where alternative locations are used for individual parts or processes, then these processes are not modelled. Each factory model contains models of all the manufacturing resources, which are either grouped into *cells*, or may belong directly to the factory.

- Cell

A cell is an administrative grouping of production equipment within the factory. Where cellular manufacturing is applied, the cell will also be a physical grouping on the shop floor. The cell contains machine tools, storage, transportation and has a labour force.

- Production Machine / Machine Tool

A machine tool, as modelled in CESS, is a device for performing a particular manufacturing process. Machine tools can be of many different types, ranging from small machines dedicated to a particular process (e.g. drill and tap), to large multi-process machining centres. Facilities such as chemical treating plants, paint shops and heat treatment ovens may also be modelled as machine tools. Each machine tool has a requirement for labour in order to work it. Some machine tools require constant attention of an operator, whilst others only require that they be set up and can load each new workpiece automatically. Machine tool models must represent the capabilities of each tool, in terms of processes which can be performed, quality and cost data and workpiece capacity. Since production machines vary considerably, depending on the processes which they are designed for, the machine tool model within the system must be specific to the class of machine tool. For example, a lathe has a different work-holding method than a milling machine; the properties defining maximum component size are length and diameter for the lathe, but length, width and height for the milling machine. Another important factor in machine tools is the production capacity, which must not be exceeded when producing a process plan.

- Work Storage / Buffers

An important part of a manufacturing plant, the work storage arrangements will depend on the production strategy. Work storage is required both within cells and at the factory level. Inventory cost (the cost of financing the work in progress and in stores) can be an important contributor to the overall cost of a product and therefore a model of the inventory is important.

- Transportation

The movement of work around the factory contributes to the manufacturing lead time and to the cost through the requirement for labour and equipment. The transportation method employed within a factory can include conveyor belts, forklift trucks, hand trucks and manual carrying. Each method has different properties of speed and cost per distance travelled. The algorithms for the calculation of both lead time and product cost should include a consideration of the transportation cost, a product of the method of transport and the distances travelled. For the purposes of aggregate process planning, it is not appropriate to model the movements of every workpiece and material handling device. Individual transportation equipment is not modelled, therefore. Instead, each factory and cell has a transportation method property which indicates the method which is used for movements within that area.

- Labour

The workforce available in the factory supply the labour. Labour is required to carry out most of the processes within a factory, and therefore it is important to know the cost of the labour. This will vary depending on the factory, since labour costs are related to the location of the factory and the hours worked. Each operation in the process plan will incur labour costs which must be calculated. CESS does not attempt to model individual workers, but instead each cell is allocated an amount of labour which can be shared amongst the jobs within that cell. This allows for machinists to operate more than one machine at once when possible.

6.3 Resource model structure

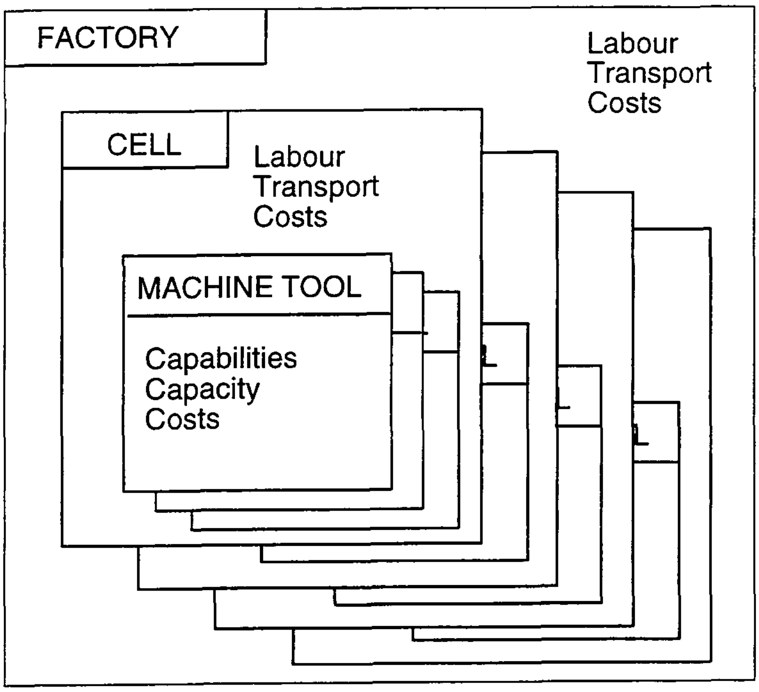


Figure 6.1: Resource model structure

The previous section has identified the elements which must be combined into the resource model. In Figure 6.1, the nested structure of a single factory is highlighted: the factory consists of a set of cells, each of which consists of a set of machine tools. In addition to the component objects, however, each level of the model has its own set of properties, representing information on, for example, the transportation system.

Whilst the factory and cell models will remain similar for most examples, the machine tool model will vary depending on the machine type. A lathe and a milling machine have different properties and therefore to model both of these with the same class, a complex and redundant model would be required. Using an object oriented model, however, specific models may be developed for each machine tool type, so that only the appropriate properties and methods are supplied for each machine tool. Each machine tool is therefore an instance of a sub-class within the hierarchy of the machine tools class (Figure 6.2).

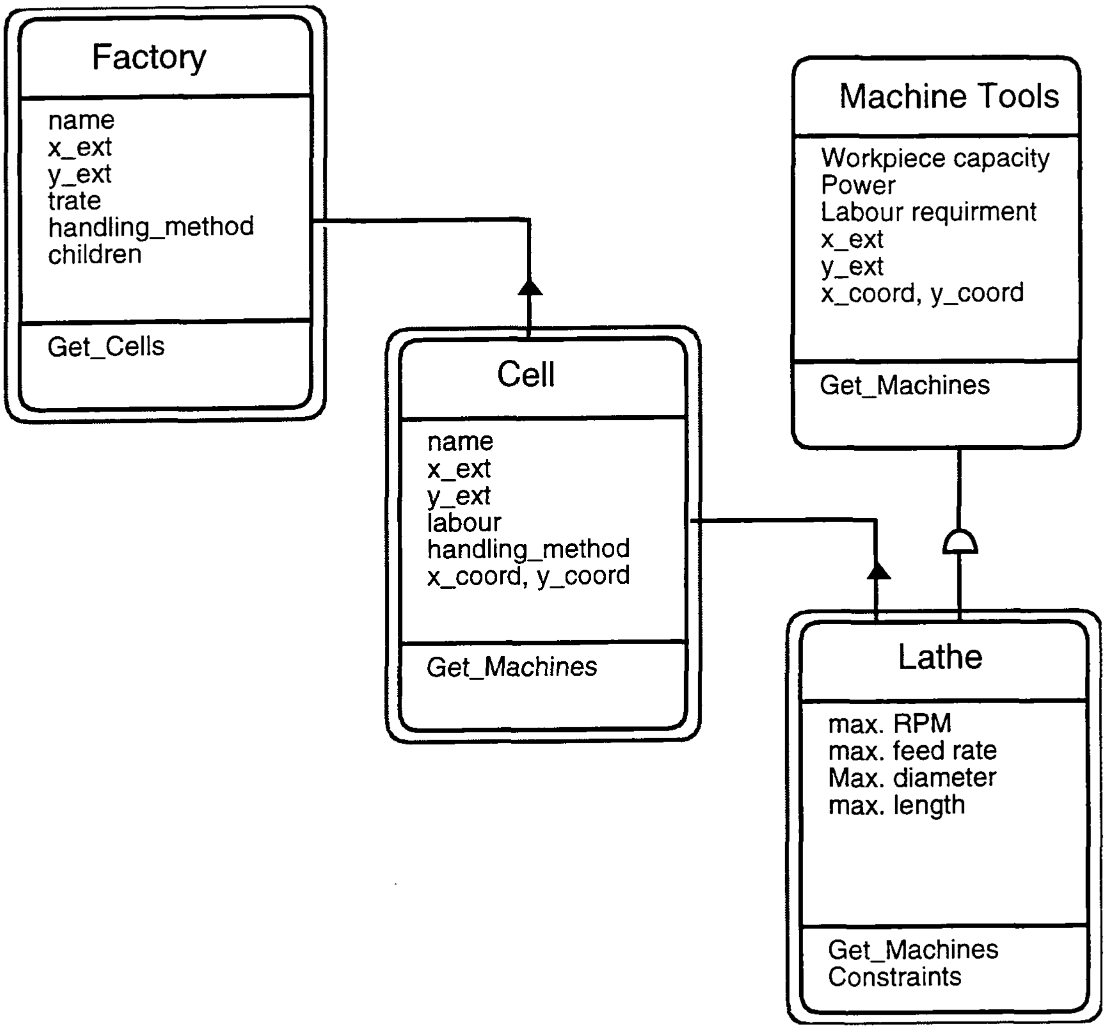


Figure 6.2: Resource class structure

6.4 Factory model implementation

As described above, the factory concept is implemented within CESS by using a single *factories* class. All factories are members of this class. The factory class has the following properties:

Table 6.1: Properties of factories class

name	A text identifier
x_ext	The width of the factory floor (in metres)
y_ext	The length of the factory floor (in metres)
trate	The cost per metre per kg of material transport.
handling_method	The material handling method
children	A list of the child objects of the factory, including any cells and machine tools.

6.5 Cell model implementation

As with the factory concept, the cell concept is implemented within CESS as a single *cells* class. It is at the cell level that the resource of labour is introduced to the model. The cell objects include positional information within the factory floor so that transportation distances can be calculated. For inter-cell transportation, the factory material handling method is assumed, whilst for intra-cell transportation, the cell material handling method will be used. The cells class has the following properties:

Table 6.2: Properties of cell class

name	A text identifier
x_ext	The cell width in metres
y_ext	The cell length in metres
(x_coord, y_coord)	The position of the cell relative to the factory floor (m)
labour	The number of full time operator/staff
handling method	The material handling method

6.6 Machine tool model

In order to define the capabilities of a particular machine tool, it is necessary to identify a set of parameters which describe it. The appropriate parameters to describe a tool will vary with the type of machine: on a lathe the critical dimension is the workpiece diameter, whilst for milling machines the dimensions length, width and height would be more important. In order to model these different types of machine tool a separate class is defined for each type. A classification of machines tools has been compiled containing a detailed model of each machine tool type in its place. By classifying the different machines types into a hierarchy, it is possible to write generic models which apply to groupings within the classification, such as all lathes, and then to modify the details of these models to better represent individual variations.

Table 6.3: Properties of machine tool super-class

available	Boolean	Denotes whether the machine tool is available for use in the aggregate process plan. Allows the user to remove machines for maintenance etc.
btime	Float	Batch set-up time: Time required to set up the machine for a new job. Includes setting up fixture, programming etc.
cell	String	The name of the cell to which the machine tool belongs
freecap	Float	Free capacity: The percentage free time on the machine
machine	String	The name of the machine
maxbreadth	Float	The maximum breadth of the workpiece, equal to the maximum diameter for lathes, (mm).
maxdia	Float	The maximum diameter of the workpiece, equal to the lower of the maximum breadth and width except on lathes, (mm).
maxlength	Float	The maximum length of the workpiece (mm).
maxwidth	Float	The maximum breadth of the workpiece, equal to the maximum diameter for lathes, (mm).
model	String	The model name of the machine
power	Float	The maximum power of the machine tool (W).
rate	Float	The hourly cost rate of the machine (£/hour)
x_coord, y_coord	Float	Position of the machine in the factory (m).
x_ext	Float	Width of the machine (m).
y_ext	Float	Length of the machine (m)

For each process model within the classification, a number of parameters are defined which hold all the information which is required by the rest of CESS. Some data such as the available power of the machine tool is common to all metal cutting machine types, along with limits as to the size and weight of workpieces which may be treated. The majority of the parameters are specific to the machine type since they would not be relevant to other machine types. The machines model maintains a model of each machine available in the factory of the company. This includes data such as machine location, set-up and waiting times, machine capacity and labour requirements. The machine tool super-class models the generic attributes which are common to all machine tools, shown in Table 6.3. These are the properties which are used by the aggregate process planning function, irrespective of the machine tool and process selected.

6.6.1 Machine tool class structure

In building a taxonomy of machine tool types, the aim was to develop a hierarchy which would allow the definition of processing capability for machines through the identification of a particular machine tool class for each process. In other words, the system should define all the machine tool types which can be used for a particular process as sub-classes of a single super-class and no machine tool types which cannot perform the process should belong to this super-class. Where this is infeasible, provision has been made within the methodology for multiple machine types to be assigned to a process.

This requirement naturally results in a machine tool taxonomy which is similar to the process taxonomy. The chief difficulty with building this taxonomy lies with multi-purpose machine tools, which have the capability of performing quite separate processes. Multi-process machine tools offer the process planner greater flexibility whilst significantly reducing the number of set-ups and the amount of transportation required for production. Consequently they are extremely valuable and popular machines. Such machine tools can fit into more than one category in the classification.

Another difficulty in the classification of machine tools is the concept of accessories. The modular design of many machine tools means that typically a basic machine tool type can be improved by the addition of a number of devices. These can often be added in any combination, presenting the problem of proliferating numbers of machines types which must be modelled individually. To overcome this problem, the concept of multiple inheritance has been used in the CESS model. In this method, a given machine tool is permitted to be a member of multiple machine tool classes. Thus, a lathe with a milling attachment would belong to both the lathes and milling machine classes. To allow this, the models of each machine tool type must be compatible, and certain rules must be added to control the inheritance of properties shared by the parent classes. Thus the CESS model allows individual machine tools to be defined as instances of more than one machine tool class, without defining explicit classes for the results of these combined machines. This allows flexibility in machine tool modelling whilst reducing the requirement for stored complexity, embodied by the number of classes in the model.

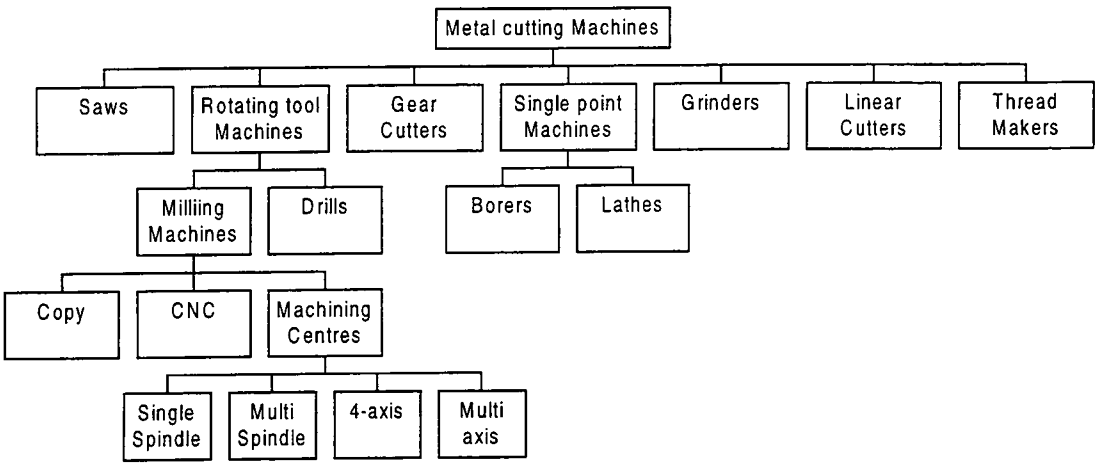


Figure 6.3: Machine tool classification

Figure 6.3 shows the top level of the machine tool taxonomy. Note that milling machines and drills are grouped as sub-classes of the same super-class (rotating tools). This reflects the ability of milling machine tools to perform drilling processes. By defining the machine type for drilling as *rotating tool machines,* and that for milling as *milling machines*, the system can allow milling machines to be used for drilling, but not *vice versa.* A similar structure is used for boring using lathes and vertical boring machines. Each of these classes is divided into further sub-classes to model individual process capabilities, as described in the following sections.

6.6.2 Rotating tool machines

This category of tools is divided into milling machines and drilling machines, as shown in Figure 6.4. The term milling machines is used in this thesis to refer to all machines capable of performing milling operations. There are a number of different types of such machine available. The main distinction is between milling machines proper and machining centres. The latter describes machine tools which have the ability to automatically change cutting tools between a magazine and the spindle so that multiple process types can be performed in the same set-up. Simple milling machines have a single machine tool loaded and require resetting to perform a second process. The machining centre is by far the more common in modern factories. Machining centres can be characterised by the number of axes of movement, with more axes allowing greater complexity of geometry to be produced. Drilling machines are effectively single axis milling machines, requiring that the tool only be moved along its axis during cutting. The different classes represent alternative configurations of tooling, layout and control.

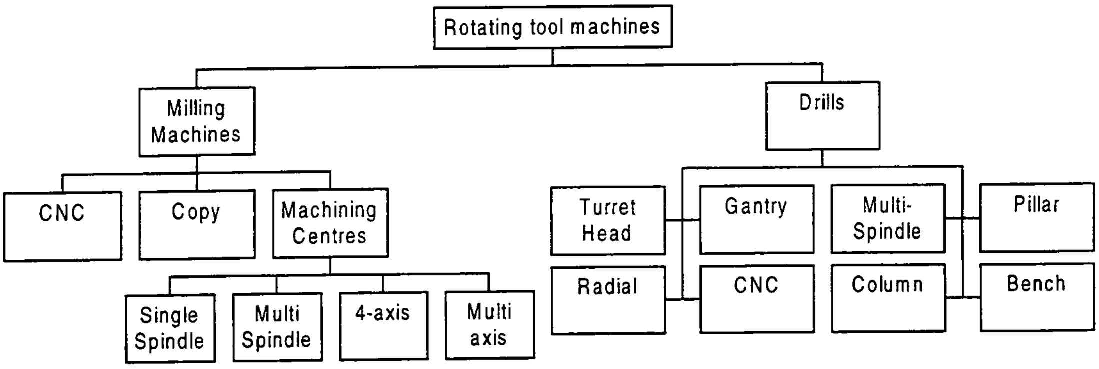


Figure 6.4: Rotating tool machine classification

6.6.3 Rotating work machines

The single point classification includes two types of machine tool, borers and lathes. In both cases the relative motion of the tool and work is the same, but in the former case the tool is rotated whilst in the latter case it is the work which rotates. The boring machine is designed for large components which are too heavy to rotate. The cutting tool is mounted eccentrically from the axis of rotation to mimic the processes of the lathe.

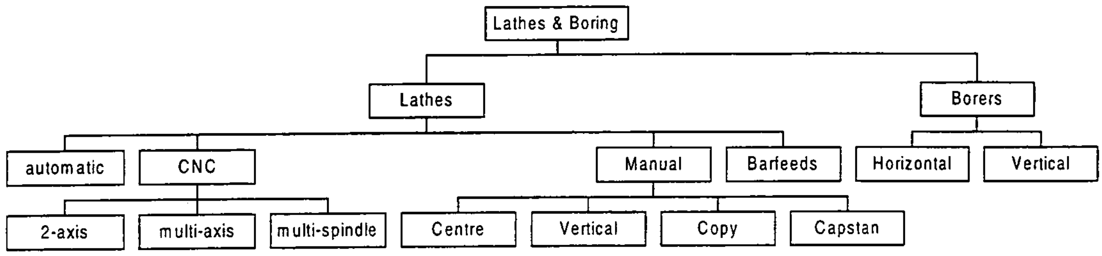


Figure 6.5: Single point machine classification

A simple lathe consists of a spindle, comprising a work-holding device (or chuck) which is connected to a motor, which rotates the workpiece about its axis. A turret, which can be driven parallel to the axis of the lathe, is used to hold turning tools. The turret may be used to position the tool at an offset from the lathe axis. In addition to this basic arrangement, there are numerous accessories designed for lathes: a tailstock may be used to support the free end of the workpiece; a drill may replace the tailstock to create axi-symmetric holes in the free end; a second spindle may replace the tailstock to allow machining of the other end of the workpiece; multiple turrets may be used at once; turrets may have powered tooling to perform milling processes and dedicated loading equipment may be present (e.g. a bar feed).

6.6.3.1 Example Machine tool model: Powered turret CNC lathe

The powered turret CNC lathe is typical of modern machine tools: it is capable of multiple processes (both turning and milling), can be modified through the addition of a number of devices (e.g. barfeed) and can select from a number of cutting tools to perform multiple operations in the same set-up. If a specific model of this lathe type was made using a single class and all other combinations were modelled similarly, then the number of classes required would become unmanageable. In the CESS model, therefore, the lathe model is built up by combining the models of the individual features through multiple inheritance. Thus the lathe is represented as an object (Figure 6.6) which is an instance of the classes of (i) CNC lathes, (ii) Powered turret lathes and in this example, (iii) Barfeed lathes.

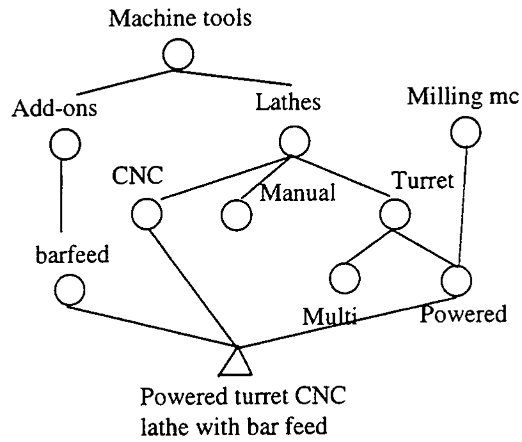


Figure 6.6: Multiple-inheritance used to specify a complex machine tool

Using the above structure for the model of the lathe facilitates the definition of process capabilities. The processes class definitions specify a particular class of machines which is required to perform the process. Thus, simple turning processes have a requirement for a machine tool belonging to the lathes class. This means that any machine which belongs to the lathes class or one of its sub-classes can perform the turning process. More complex or specialised turning processes would have a particular sub-class of lathes defined. Thus, profile turning, which requires dynamic control process parameters to vary depth of cut continuously, have a requirement for CNC lathes. Similarly, milling processes have a general requirement for a milling machine or machining centre. In this case, the powered turret class of the lathe is also defined as a sub-class of the general milling class. This means that the lathe automatically inherits the capability to perform both milling and turning processes as it is a member of both classes. Certain milling processes can be excluded from the list of capable processes, of course, if they have a requirement for a specific type of milling machine.

6.6.4 Grinding machines

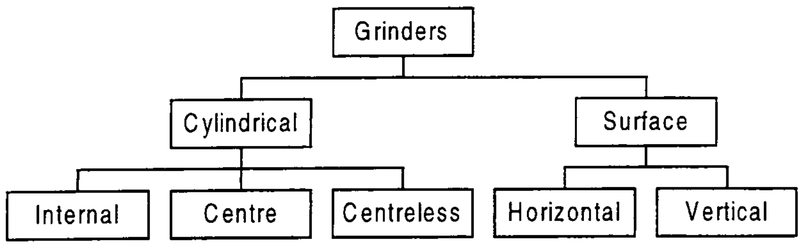


Figure 6.7: Grinding machine classes

The CESS classification of grinding machines (Figure 6.7) is divided into two sections, surface and cylindrical grinders, according to the surface which is generated. Surface grinders produce flat surfaces, through traversing an abrasive wheel across the workpiece with its axis either parallel to the work surface (horizontal) or normal to the surface (vertical). In the former case the cylindrical surface of the abrasive wheel is the cutting surface, in the latter it is the flat surface of the wheel which cuts the workpiece. Cylindrical grinding produces either internal or external cylindrical surfaces, by rotating an abrasive wheel in contact with the rotating workpiece. Within this classification there are minor modifications to the machine depending on whether the workpiece is held in a chuck or between supporting wheels. The latter machines (centre-less grinders) are designed for larger workpieces. In general, however, the model for all cylindrical grinding machines is the same.

6.6.5 Miscellaneous machine types

In Figure 6.8 the a number of additional machine tool classes are specified, in addition to those mentioned thus far. These machine tools are those which can perform the less common, or minor manufacturing processes. Four classes are identified:

1. Linear cutters includes machines which remove material by driving a solid tool through the workpiece in a straight line. Shapers and planers are single point cutters, whereas broaching tools have multiple cutting edges. These processes have generally been rendered obsolete by new processing methods.

2. Gear cutting machines are specialist machines designed to produce a variety of gear designs. Gear making has not been modelled in this project so this branch of the machine taxonomy has not been further expanded.

3. Thread making machines are used either for die threading or for tapping. These are specialist machines which are not generally used since forming the threads on either drills, lathes or milling machines is often more economical and convenient.

4. Saws or sawing machines are another type of machine which has not been investigated for the project. Sawing machines are designed to pass a toothed blade through the workpiece. Sawing processes are more often used as primary processes to produce billets than for shaping of components. Therefore these machines have not been modelled within CESS.

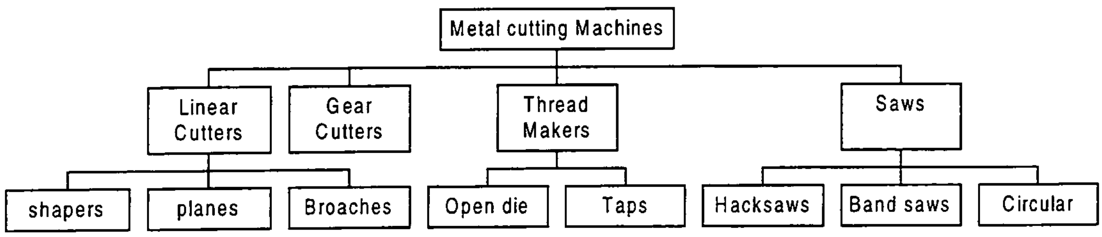


Figure 6.8: Miscellaneous machine classes

6.6.6 Additional equipment and accessories

For many machine tools, additional equipment is available to improve the overall production. In particular, several methods have been devised to reduce loading and setting up times for both milling machines and lathes. The oldest of these is the bar feed attachment for lathes, which allows raw material in the form of long bars to be used to create many products from a single piece. Milling machines and machining centres are generally available with multiple pallets to allow off-line reloading. Robots are frequently used to load single pallet machines and even lathes, picking workpieces from a pallet.

The added functionality and benefits that these accessories provide is generally modelled in CESS by defining specific classes for the accessory and making the machine tool an instance of this class. Thus the classes represent machine tools with the device rather than the device themselves. The impact of these devices is largely in reducing set-up times, but there are also often benefits in quality due to the improved accuracy of part placement. The quality analysis system will reflect this since it compares the performance of similar machines types and will make the distinction between a manually loaded and robot loaded machine.

6.7 State of the Art Model

During the development of new products, the process planner will often wish to consider the purchase of new machine tools. The best way in which to assess the impact of a new machine tool and to determine if it would be an sound economic investment is to calculate the effect on the production of new and existing products. If a model of the new machine tool is available within the process planning system then the machine tool may be considered alongside the existing equipment and properly assessed. It is proposed that this could be a useful application of CESS, since it is a relatively simple task to build models of state-of-the-art equipment by selecting the appropriate classes within which the new machine tool should be classified. If the parameter values such as speed, feed and power limits are available for a new machine, then it can be included in the factory model and the new process plans generated may be compared with previous sets.

6.8 Suppliers and Subcontracting

The decision to make or buy a particular component will depend on the availability of processes within the local factory, and the capacity of the machine tools which can perform these processes. In the future, it is anticipated that a company using CESS would use the tool to assist in the make or buy decision by comparing the costing of the component built in house with that built by known suppliers, using a model of the supplier company's resources. In order to implement this approach, a close relationship would be required between the two companies since it implies the sharing of potentially valuable information about each company. The integration of suppliers and customers into the CESS methodology and strategies for the consideration of make or buy options is the subject of further research (Darlington and Maropoulos, 1997).

At present within CESS, when a supplier is specified for a component part then the cost must be entered by the user. In some cases, the routing may require the use of an external contractor for just a single process. This is usually the case when the company does not have the capability to perform a particular process, and it would be uneconomical to purchase the necessary equipment and train staff for a single product. Typical processes requiring external contractors are material treatment processes, such as heat treatment and surface coating processes. Sometimes the contractor is a subsidiary of the company, but the factory is at a remote location. In situations such as these, the resource model represents the costs of transportation of the workpiece (typically in large batches). The processing costs of processes performed by external contractors will be replaced by the charge given by the contractor, unless that company's factory is modelled within CESS. If there is a model present for the contractor's factory, the cost can be estimated in the same way as an internal process, with the addition of a margin of profit determined by the contractor.

In today's business environment, it is quite common for large companies which buy in services or components from small manufacturers to be in a position where they can specify a profit level which the supplier is permitted, and then to cost the job and set the price level which they are willing to pay. This will happen when the customer's product is a significant (greater than 30%) proportion of the supplier's overall turnover.

6.9 Summary

A set of models has been developed to represent the resources available to the company in manufacturing the product. In combination with the process models, this allows the assessment of the manufacturing options for a given design. The system can predict the effect of machine tool selection, factory layouts and staffing levels. The use of a detailed resource model within the system should allow the addition of extra functionality such as the use of the system for performing benchmarking of the factory against the state-of-the-art, and against other sourcing options, thus allowing the company to determine which parts should be made in-house, and which should be bought in from suppliers.

Chapter 7

Aggregate process planning

7.1 Introduction

This chapter defines the concept of aggregate process planning through the description of the implementation of an aggregate process planning methodology within CESS. The CAPP functionality of CESS analyses the product model and produces a set of alternative aggregate process plans and routings using a model of the factory and the aggregate machining process model. The current implementation of this system operates at the component level, whilst the next phase of the research will incorporate the permanent assembly level through modelling of fabrication operations.

Aggregate process plans consist of a hierarchical set of instructions which can be mapped against a structured aggregate model of the product design. The aggregate process plan gives a general description of the production method for the product at feature level. Suitable combinations of processes and resources are identified and an appropriate sequence of operation is set out. An aggregate process plan is intended as a guide to indicate manufacturing options for a product and an indication of cost, lead time and quality. It is not a complete set of manufacturing instructions in that it does not include "machine code" required by machine tools.

7.1.1 Terminology

An aggregate process plan for a single component is termed a *job*. This consists of all the instructions necessary to create the component. Within the job there are definitions of the raw material source, or *blank*, and a set of instructions, the *operations*, for the generation of each design feature. A single operation describes all the processing stages required to create one feature. Since a single feature may require a number of processing steps, a further level of instruction is defined within the operation and each step of the operation is modelled as an *operation element*. A single such element defines one step in the process of creating a feature. An example of an element might be rough turning of a cylindrical surface. The element defines the quality level at which a process is specified, along with the machine tool to be used and the amount of material to be removed. The sum of all the material removal for the elements of one feature will create the feature. The structure of an aggregate process plan is illustrated in Figure 7.1.

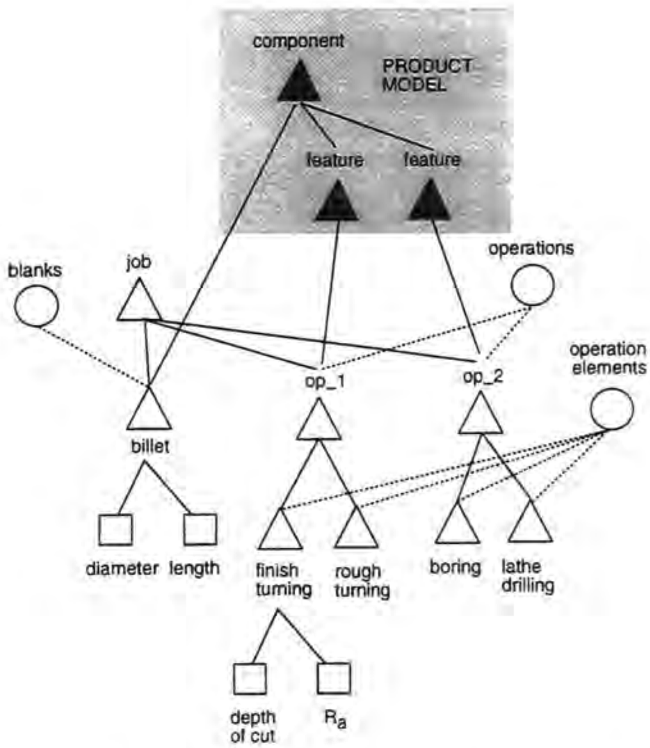


Figure 7.1: The structure of an aggregate process plan model

7.2 Specification of the aggregate process planning function

There are a number of requirements which an aggregate process planning algorithm must meet. The specifications have been used in order to develop the functionality described herein. A key consideration is that aggregate process planning is a generic technology which is intended to be applied across the full range of production processes to allow the comparison of all manufacturing options for any product. The other requirements for aggregate process planning are listed below:

Early design	The algorithm must be able to operate on early design data, i.e. at the conceptual and embodiment stages where much of the detail required by traditional CAPP systems is not available.
Variable detail	Aggregate designs will vary in detail from component to component, and within the same component over time. So the system must account for this variation and use extra detail where available.
Non-linearity	Aggregate process planning should identify a range of alternative routes, with comparative evaluations, rather than homing in on a single option.
Process identification	The algorithm must identify the processes which could be used for manufacture of the design. A key feature of aggregate planning is the consideration of a wide range of processes.
Resource selection	Process plans must involve the specification of resources including machine tools. The plan should determine the production capacity required for each resource.
Sequencing	The process plan will involve a series of steps, which must be organised into a logical sequence. Early knowledge of production sequence enables the planning of facility layout and schedules to be brought forward in time.

Transparency	The algorithm should provide clear feedback to the user when decisions are made to ensure the reasoning is understood and so alternative options are not discarded out of hand.
Customisation	Provision must be made to allow the user to influence the decisions made by the algorithm to reflect outside influences such as company business strategy.
Multi-criteria analysis	The optimisation within the algorithm must reflect the multiple criteria which must be satisfied to arrive at a good process plan. These include cost, quality and lead time.
Realism	Clearly the process plans produced by the algorithm must be in line with those which would be adopted within the company so that they provide a reasonable guide to expected final production costs.

From the above objectives, it is possible to fashion a functional description of the routing algorithm. The algorithm must analyse each feature of the product design at the detail level specified, and form a list of possible manufacturing processes. For these processes the required resources must be identified and then the system must determine suitable combinations of process and resources, in a specified sequence, such that the criteria of cost, quality and lead time are optimised. The system should produce a number of alternative production plans at the same detail level which can be compared. At each stage of the algorithm where important selections are made, the user should be able to view the alternatives which were available and the choices which the system made. The three main tasks of route generation are detailed in the following sections.

7.2.1 Process selection

The basic principle of process selection is that for any given geometry of product, there are one or more processes which can be used to produce it. Where only one process is possible for a particular feature then this function is simply a requirement for that knowledge to be captured within the knowledge base of the system. More usually, however, there will be a variety of processes which could be used; in some cases variations between processes are small, whereas in other cases the process works on very different principles. Where a choice of two or more processes is available for a feature, the system must not only identify this fact, but must make an informed decision of which process to select. Selection between alternative process types should be performed on the basis of the cost and quality of the product. If the cost of a particular quality level can be calculated, then the criteria for process selection can be reduced to a single indictor, total product cost. Therefore, the system has cost models for each process type, which will calculate the cost of using that process for a given feature and which will include the costs due to the quality level of the process.

It is important to note, however, that when considering the selection of a process at a feature level, each process which is used will generally require the use of a separate machine tool or system. Each machine tool used will generally incur the addition of extra costs for setting up, and for transfer of the part between workstations. This is why the "design for manufacture" philosophy specifies that the number of different process types required for a product should be minimised. Therefore, the costs of use of a particular process for a feature cannot be calculated in isolation for just that feature. In order to accurately reflect the true cost of using a process, it is necessary to consider which other processes are being used and whether the current process is used for other features on the component.

It can be seen that to develop an accurate model of the processing cost for a feature requires information about the resources that are to be used: metal cutting processes can be performed more quickly and therefore more cheaply on machine tools with higher power. However, if the process selection is performed before the resource selection, this information on the resources will not be available.

7.2.2 Resource selection

Resource selection refers to the specification of which machine tools and equipment will be used to carry out the manufacturing process. Each machine tool can perform a particular set of processes, typically from within the same overall classification. For example, vertical milling machines can be used for milling and drilling operations. The choice of machine tool will determine exactly what process parameters are to be used and therefore will effect both the cost and the quality of the product. The criteria for the selection of machine tools include; the volumetric capacity of the machine, the accuracy of the process as carried out on that machine, the speed of the machine (both travel and spindle speeds), the cost rate of the machine and the location of the machine within the factory. In many cases, the planner will wish to select machines such that the component is made fully within one cell in the factory.

As for the process selection, the prime reason for the selection of one machine over another is the cost. The process planner must therefore identify the machines which are suitable for the manufacture of each feature, and then select the best machine or set of machines, based on an analysis of the costs of manufacture. Once again, it is important to note that the costs of machines can be spread over more than one feature, and therefore it is only possible to assess costs if all features are considered together.

With resource selection it is particularly important to provide comparative information between the different machines in the output from the system. It is often necessary within lower volume manufacturers to move production originally planned for one machine tool to another one. This can occur for maintenance reasons, or because of a lack of capacity due to scheduling difficulties. CESS can be valuable in this situation since alternative production plans can be considered during the development and the cost of using an alternative machine can be clearly determined.

7.2.3 Task sequencing

The third requirement for the generation of a working process plan is that the planner must specify the order in which the basic tasks are to be carried out. There are many influences on the order in which jobs will be carried out, some of which may be set aside, whilst others cannot be altered. Amongst these influences are: process type, feature type, quality constraints, geometrical constraints and ergonomic constraints.

The effect of the sequence is to apply constraints on the selection of processes and machine tools. In order to minimise costs it is important to minimise the number of set-ups and the amount of transportation involved in a production route. If the sequence of tasks is chosen incorrectly then the number of set-ups and the amount of travel will be increased unnecessarily. A typical example of this approach would be to plan all elements of machining to be clustered according to the feature. If a separate roughing and finishing process were then to be used, the component would have to travel from one machine to another once for each feature. On the other hand, if the elements of the plan are clustered by quality level or process, then only one transfer would be required, after all the roughing had been carried out.

7.3 Aggregate process planning functionality: Overview

The previous section set out the requirements for the process planning logic within CESS. In this section the structure of the proposed process planning algorithm will be detailed. The architecture which has been adopted consists of a two stage selection process whereby the process type is chosen first using general machine data, whilst the choice of a specific machine tool is made afterwards, using the data specific to each machine. The sequencing function is carried out in between these two stages where it can be most accurately applied. In both cases a list of options is generated for each feature and then a genetic algorithm search technique is used to find the best sets of solutions based on the criterion of minimum cost.

The decision to perform the three tasks of process selection, machine tool selection and sequencing sequentially instead of concurrently was taken in order to reduce the computational load on the system. It is important for an aggregate process planning system to operate rapidly in order to provide immediate feedback to product developers, particularly when used by designers. This allows the evaluation of many alternative ideas and this is the key to successful conceptual design. Whilst it is generally a straightforward procedure to generate the lists of alternative production options which are available on a feature by feature basis, finding the best combination of each of these options is a more difficult task: The size of the search space increases exponentially with component complexity and the number of available machines. Indeed, for most practical problems, the search space becomes too large to be effectively searched with even advanced methods, and therefore it is necessary to adopt heuristics to reduce the number of options to manageable proportions.

For this algorithm, it has been decided that the effective way to reduce the number of route possibilities that should be searched whilst retaining the greatest chance of reaching the optimum solution is to remove the sequencing of the manufacturing options from the optimisation stage. By determining the sequence of the manufacturing operations in advance of machine selection it is possible to reduce the number of possible manufacturing options greatly. The sequence is the factor which is most dependent on engineering knowledge and expertise, and is therefore the least suitable for automation within a computer system. It is, therefore, appropriate to use a knowledge based computer system for this section of the algorithm. This technique allows the embodiment of human engineering knowledge and the easy integration of extra requirements and constraints from the user. The sequence is therefore determined with a purely heuristic algorithm that is based on accepted engineering practice and geometrical information about the component. The user may alter the sequence which is generated before allowing the system to perform the search for the optimum machines.

The aggregate process planning functions are implemented in an algorithm which divides the planning tasks into a sequence of discrete stages at which the user is consulted and is able to monitor the system's progress. This approach allows the user to develop an awareness of the tasks involved in process planning and understand the effect of each element of the design upon the production plan. It also gives the user the opportunity to override the computer generated suggestions when special circumstances dictate. At each stage the user can choose from a number of alternatives, either allowing the algorithm to consider a wide range of options, or narrowing the field down by the application of further constraints. The outputs of the selection procedures are lists of alternative options, which are sorted according to cost and time, from which the user may choose the most desirable based on specific criteria. The overall structure of the aggregate process planning module of CESS is shown in Figure 7.2.

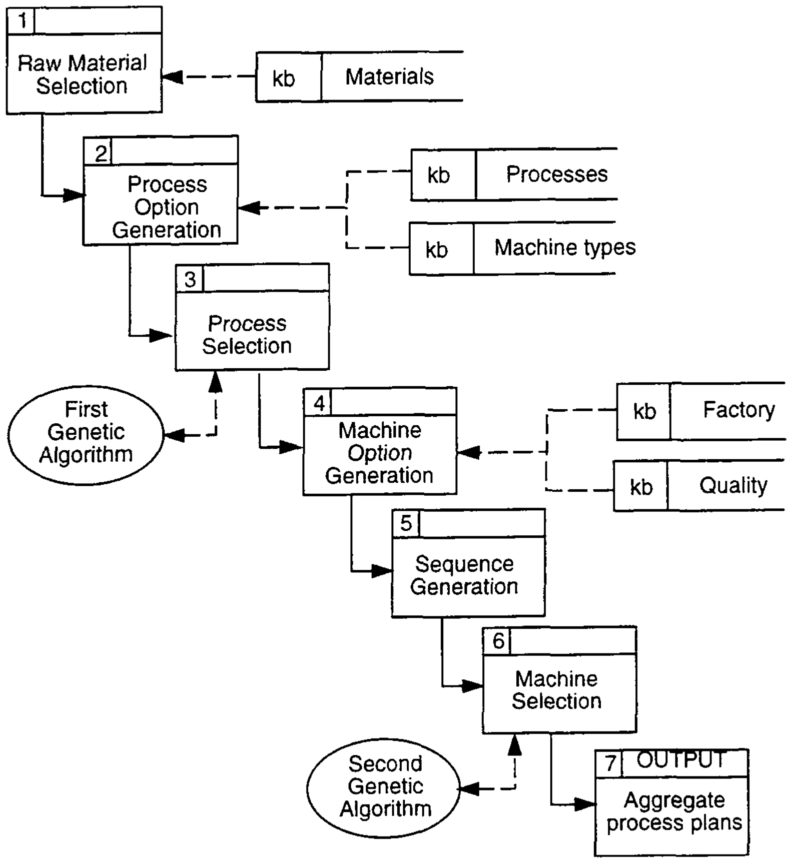


Figure 7.2: Overall aggregate process plan structure

7.4 Raw material selection functionality

The process planning function of CESS will operate on each of the product model features to generate a manufacturing route consisting of discrete operations. These operations may consist of one or more steps which form the feature shape from the initial condition. In order to determine the number of steps required, the processing time and the suitability of certain processes, it is necessary to know the initial state of the material from which the feature is to be created. For early design modelling, however, it is desirable to avoid unnecessary constraints on the raw material. Designers should be given as much freedom as possible; whilst the manufacturability must be considered, the normal processing route should not overly constrain the designer. In addition, the designer should not constrain the process planner by basing the product model on a less than optimum raw material shape.

In summary, therefore, CESS requires a means of modelling the initial condition of the workpiece at the point where each feature is placed. This model should provide for the rest of the part to be unspecified if desired. The model should store enough information to determine the amount of material removal required for the feature using aggregate process costing methods. In addition, CESS requires a functional module to provide the information for this model of the raw material condition of the part. This module must build the raw material or *blank* model during the creation of the product model. Data for the blank model is supplied through a combination of dialogue with the user and interpretation of the product model.

7.4.1 Blank Model

The blank model defines the initial geometry of the material which becomes the component. The blank model is created only during the routing procedure and forms a part of the route model. Blank information is stored in two classes of object: the *blank* object and in *operation elements*. The blank object stores a representation of the initial overall shape of the workpiece, whilst the operation elements store information about the initial conditions for each processing step. This data is calculated from the blank dimensions and the depth of any previous cuts.

The blank object will be an instance of one class of the *blank types* classification. Blank types are classified according to geometry, and include the commonly available raw material shapes, together with classes to represent pre-shaped components such as castings and forgings (Figure 7.3). The geometry of these complex shapes need not be fully represented for aggregate planning purposes: the aggregate process planning function only requires that the additional geometry which is to be made using machining processes be defined in the product model.

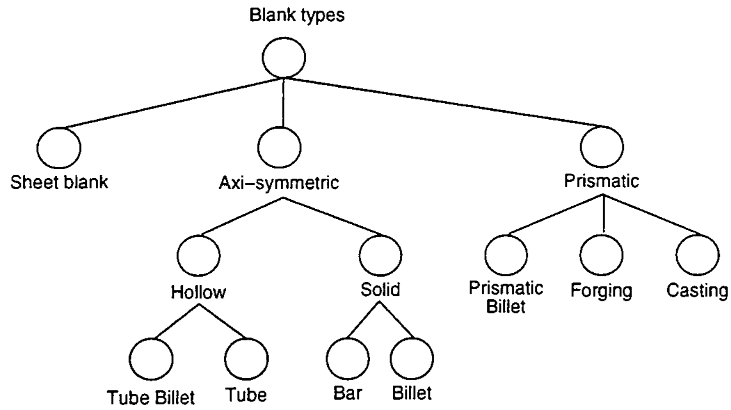


Figure 7.3: Blank type classification

Each blank type sub-class has a specific set of properties, which define its geometry. In addition, all blank types inherit two properties from the super-class: the Boolean property *rough* and the dimensional property *oversize*. The rough property is used to indicate whether or not the features are partially formed on the blank: if the property is set to TRUE, then the individual features are considered to be already present in rough form on the blank. This is typically the case for forgings and castings, which are formed to near final shapes. The oversize property is used to define a default value for the amount by which the initial workpiece dimensions vary from the component design dimensions as specified by the features. This property is a value in millimetres which the dimensioning algorithm can apply to the positive feature dimensions to get an approximate size for the blank. In addition, it is this value which is added to the final feature dimensions to determine the initial size when the blank is specified as near to final shape.

7.4.2 Pre-processing of Raw Materials

In some cases pre-processing of the raw material will be carried out at the factory, such as painting, cutting to approximate size (billeting), shot-blasting and degreasing. These processes are used to improve the performance of later processes, and may be considered as roughing processes for several features at once. The implementation of CESS described here does not model the use of pre-processing operations although it is recognised that a fully implemented system should do so.

7.4.3 Blank selection

Definition of the blank is a three stage process: First the blank type must be determined, then any preliminary forming for each feature must be assigned and finally the depth of cut for each feature can be calculated.

The first user input during the aggregate process planning function is a selection of the blank type to be used. This selection of blank type is made from a list of options that is generated from two sources: the shape of the component (determined from the positive feature) and the material type. Each positive feature class has a pre-defined list of permissible/suitable blank types (shown in Figure 7.4). Similarly, each material type stored in the materials database has a list of available blank types. Both these list are sub-sets of the full list of blank types shown in Figure 7.3.

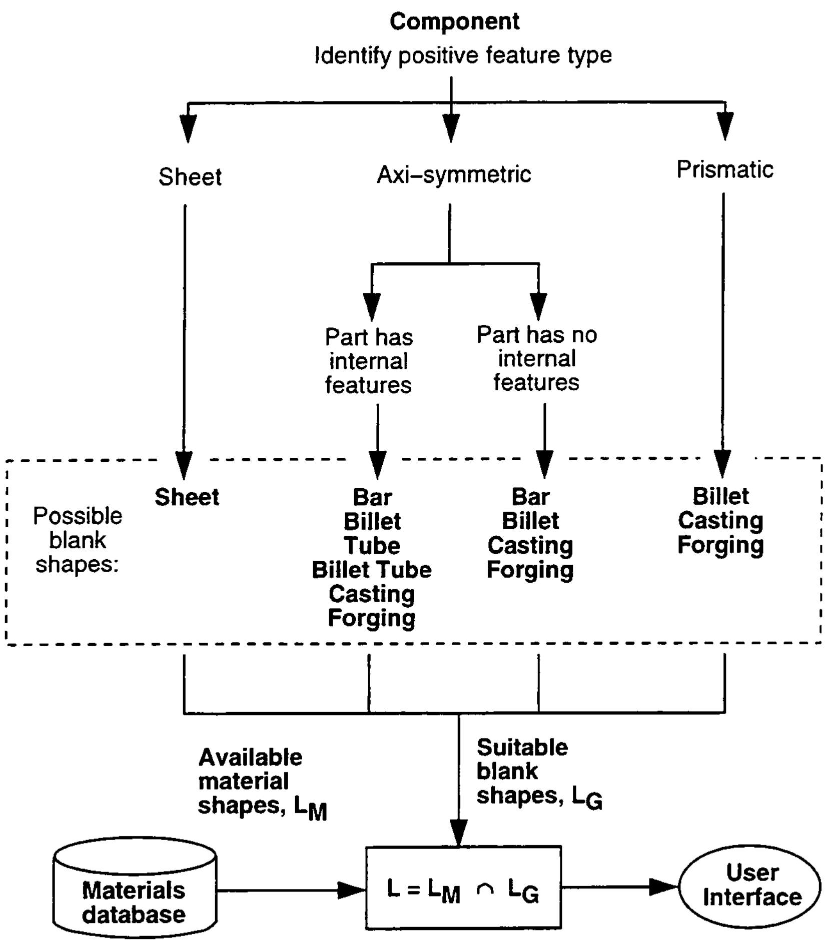


Figure 7.4: Blank type selection schematic

Three sets of suitable blank types are defined depending on the positive feature. If the positive feature is a sheet feature then the blank shape must also be a sheet. If the feature is prismatic, then three blank types are suitable, billet, casting and forging. In general the user would select billet unless a primary forming process was planned. If an axi-symmetric positive feature is selected, however, the system does not select from a simple list. Instead, a manufacturing knowledge rule is applied to suggest a refined list. If the component has no internal features, then the user must select from bar, billet, casting and forging. If the component does have internal features, then the blank types of tube and tube billet are also suggested. These types are clearly unsuitable for solid components, however they can save considerable machining time for hollow parts.

The material blank type set is retrieved from the materials database. The blank types listed for a particular material depend on the availability of the material in different raw forms and on whether the material can be shaped using the casting and forging processes. To determine the complete list of possible blank types, the system finds the intersection of these sets. This list is then passed to the user who is prompted to select the desired blank type from the allowed list. Once a blank type has been determined, the system creates an object belonging to that class and to the component object. The dimensions of the blank are generated automatically according to a method specific to each blank type: each dimension is assigned to be slightly larger than the dimensions of the positive feature.

Once a blank type is known, the initial conditions for each component feature must be identified at this stage. Features are analysed to determine whether they will exist on the blank in rough form, or whether they must be machined "from scratch". Some examples are discussed below:

- External Feature on Cylinder

  For an external feature on a cylinder, if the blank type is a billet, a bar or a tube, then the blank diameter will be equal to that of the positive feature (the cylinder). If the billet is a forging or a casting, the blank diameter will be equal to the final diameter plus the minimum machining depth. This is the amount of material which is left for machining to allow the process to create a good shape and remove the effects on the material of the harsh primary process.

- Internal Feature on Cylinder

For an internal feature on a cylinder, if the blank type is billet, bar, forging or casting, then the blank internal diameter will be equal to zero (solid). This will force the use of a roughing process for access reasons. If the blank type is hollow billet or tube, then the blank internal diameter will be the internal diameter of the feature less the minimum machining depth (q.v.).

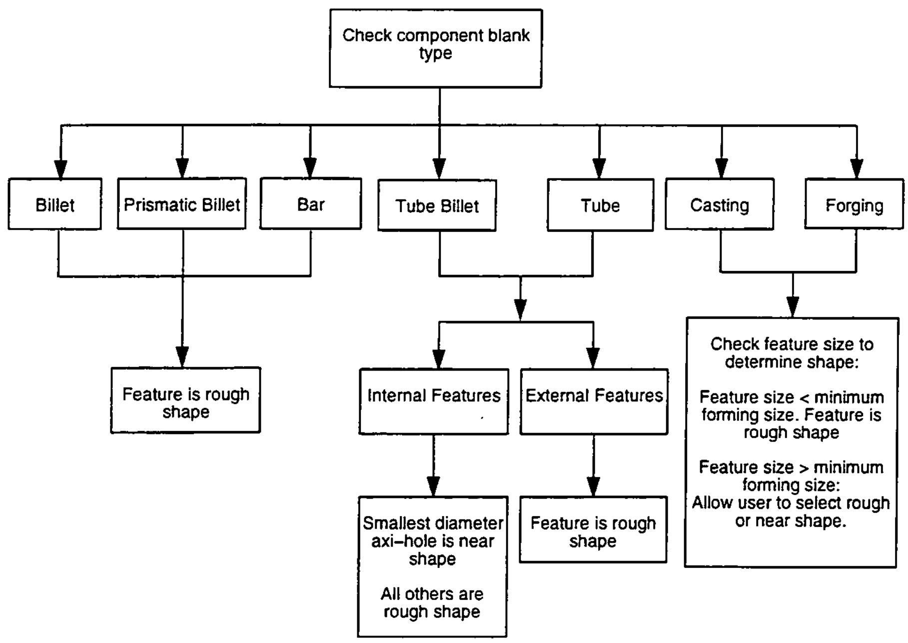


Figure 7.5: Blank detailing - Feature status assignment

In some cases the feature will be partly formed in the blank and is therefore near to its final shape (or near shape). Other features have no presence on the blank and so the whole feature must be removed. These features are called "rough shape" features. The system uses a set of rules to determine which category each feature falls into. In the case of castings and forgings, there are two approaches, depending on how much control the user wishes to retain. In the first case, the user is allowed to select which features start in which condition, subject to process capability limits. In the second approach, the size of each feature is related to a transition value which is defined for each blank type. If the feature size is greater than this transition size, the feature is assigned to be partly formed in the blank, i.e. it is "near shape". If the feature size is less than the transition size, the feature cannot be made using the process which creates the blank and so it is assumed to be "rough shape". Figure 7.5 shows the logic which determines the initial status of the feature.

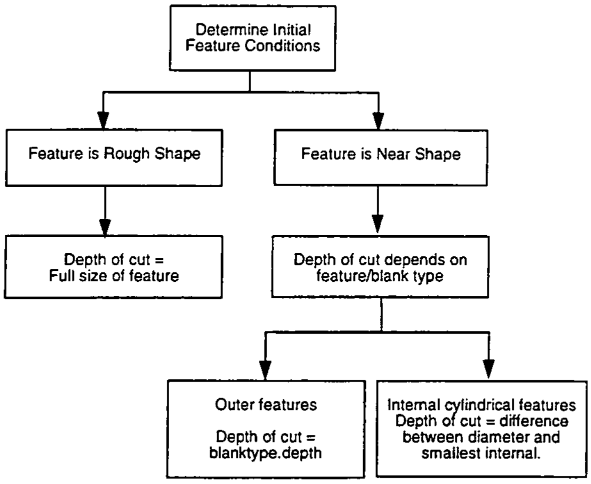


Figure 7.6: Blank detailing - Feature dimensioning

In the final stage of blank modelling, the system calculates the depth of cut for each feature. The depth of cut for a feature depends heavily on whether a feature is "near shape" or "rough shape". Figure 7.6 shows the dimensioning rules applied once the initial status has been determined. If the feature is "near shape" the depth of cut will be equal to the depth of the feature plus the oversize amount of the blank. The dimension used to calculate the depth of the feature depends on the feature type: for external and internal cylindrical features the depth of cut is the difference between feature diameter and blank diameter. For face cylindrical features, the depth of cut is determined by the length of the feature. In the case of prismatic features, the depth of the feature is used to determine the depth of cut.

7.5 Process Option Generation

This function produces a list of alternative manufacturing processes for each of the features of the component. The processes are assessed against quality constraints to determine suitability and a path from rough processing through to finishing is generated for each finishing process type. Thus, each process alternative may involve multiple processing stages. Each of the process alternatives is analysed according to the criteria of cost and production quality. The process generation algorithm utilises a knowledge base of processing types which is linked to the definition of feature types by the feature:process matrix, shown in Appendix A. The operation generation algorithm can be divided into four stages; (i) generate the list of possible finishing processes, (ii) determine quality levels for each process alternative, (iii) determine any roughing steps required for each alternative and (iv) evaluate manufacturability indicators for each option (i.e. cost, time, quality). Figure 7.7 shows the above procedure.

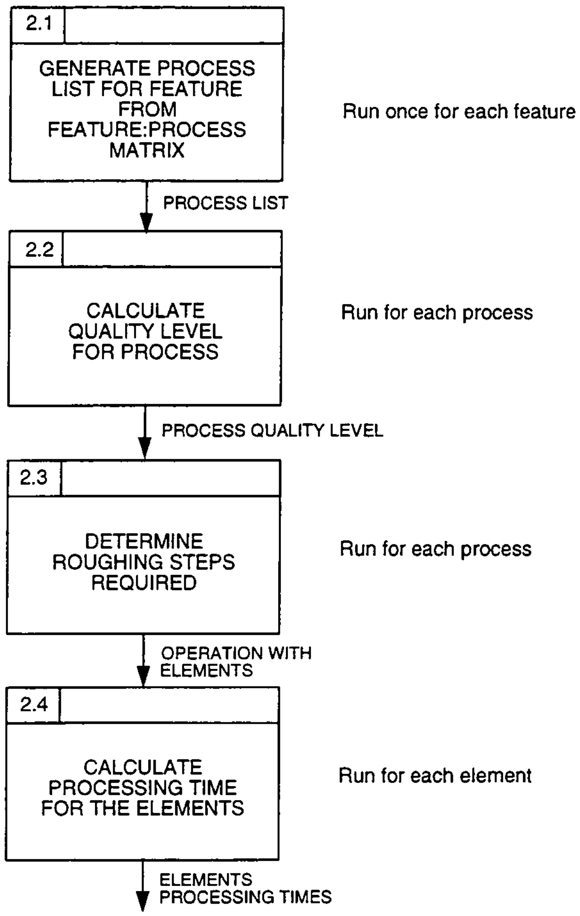


Figure 7.7: Process option functionality

7.5.1 *Identifying possible processes*

In the first stage, a list of candidate finishing processes is retrieved from the feature:process matrix. Initially, therefore, this process list is based solely on the geometry of the feature, thus the processes must be checked for other factors. This is achieved through the application of constraints. The constraints applied at this time can be defined for process or for feature. Two important constraints are considered by CESS at this stage: Material and Resource availability. To reflect the applicability of processes to certain materials, each process has a method defining a list of material types for which it is not suitable. Where the process and material are incompatible, the process is removed from consideration. Resource availability is a less generic constraint, which can be applied if required. This reflects the fact that often a manufacturer will wish to make a product without investing in new equipment or processes. This constraint allows the system to reject processes which require equipment which is not currently available in the factory. It is, therefore, the intention for the system that this is an optional constraint, permitting the system to be used for assessing potential new purchases.

7.5.2 *Determining quality levels required for each process*

The algorithm to determine the appropriate quality level of a process for finishing the feature is shown in Figure 7.8. The principle which is applied is to use the lowest quality level of a process possible whilst achieving design quality. Since higher quality levels are more expensive to attain, this approach should minimise costs. If none of the available quality levels can achieve the design quality the process is unsuitable for finishing and should be discounted. The implementation of this method uses an iterative technique. The list of possible quality levels is constructed for the process, ranked in order of lowest quality first. If there are q distinct quality levels for the process, the list can be expressed as *process[q]*. There may be between one and four quality levels, depending on the process.

In the figure, the current process quality level being evaluated is labelled as Q_p. The quality level has properties detailing the best tolerance grade achievable ($Q_p.bestIT$) and the best surface finish achievable ($Q_p.R_a min$). The feature to be machined as requirements for a certain tolerance grade (*feature.IT*) and a certain surface finish (*feature.R_a*). The algorithm checks to see whether both these requirements are met by the current quality level. If this is true, then the algorithm terminates, selecting the current quality level. If the quality requirements are not met, then the algorithm will select the next quality level in the table, if there is one. The current level Q_p is moved down the list *process[]*, until a valid level is found or there are no more entries. In the latter case, the process is rejected as unsuitable and its corresponding operation is removed from the route model.

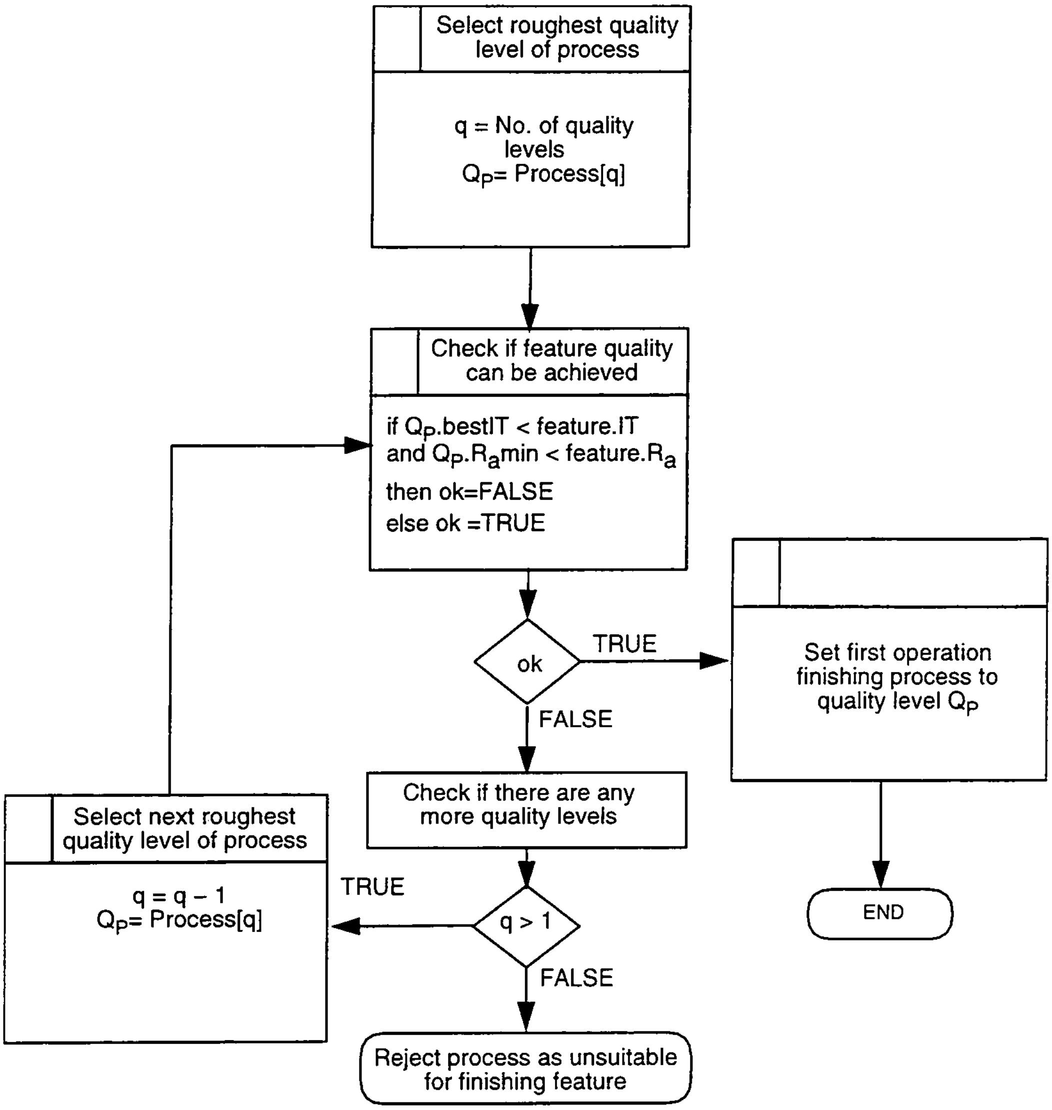


Figure 7.8: Initial process quality level schematic

7.5.3 Detailing processes and creating roughing steps

The third stage of the algorithm is to determine processing details and find out how many process steps are required. Now that the quality level requirement for the finishing process has been established, the system can create the route objects within the aggregate model. An *operation element* is created to store the overall processing information for the feature. The operation element objects belong to the process and quality level classes which they represent as well as the operation. Each operation must be analysed to determine if any roughing elements are required before the finishing process. If this is the case, the amount of material removal for each operation element must be determined.

The algorithm which is used to determine roughing requirements and to detail each of these stages is shown in Figure 7.9. Each process quality level has pre-defined limits to the amount of material which it is suitable for removing, as well as the initial conditions it requires. High quality finishing processes are designed to remove only small quantities of material, so if the feature is large, roughing will be required. The purpose of the algorithm is to compile a list of operation elements which is capable of taking the feature from the initial blank conditions to the completed design requirement. To achieve this, the algorithm starts from the finishing element and creates additional elements where required until sufficient material removal has been planned. In Figure 7.9, the element which the system is current focusing on is represented as E_c. This element has four properties, of which the process (E_c.*process*) shows the current process which is used and the quality. The other three properties, the depth of cut (E_c.*doc*), the surface finish ($E_c.R_a$) and the tolerance grade (E_c.*IT*) are the initial conditions for the operation element. These requirements are evaluated against the limits of the process quality level to determine whether further roughing is needed. The main limitation for the processes is depth of cut, but the initial tolerance level and surface finish can be important for high quality processes. For example, finish grinding would not be suitable to machine a feature if the initial surface finish was very poor, even if the depth of cut required was minimal.

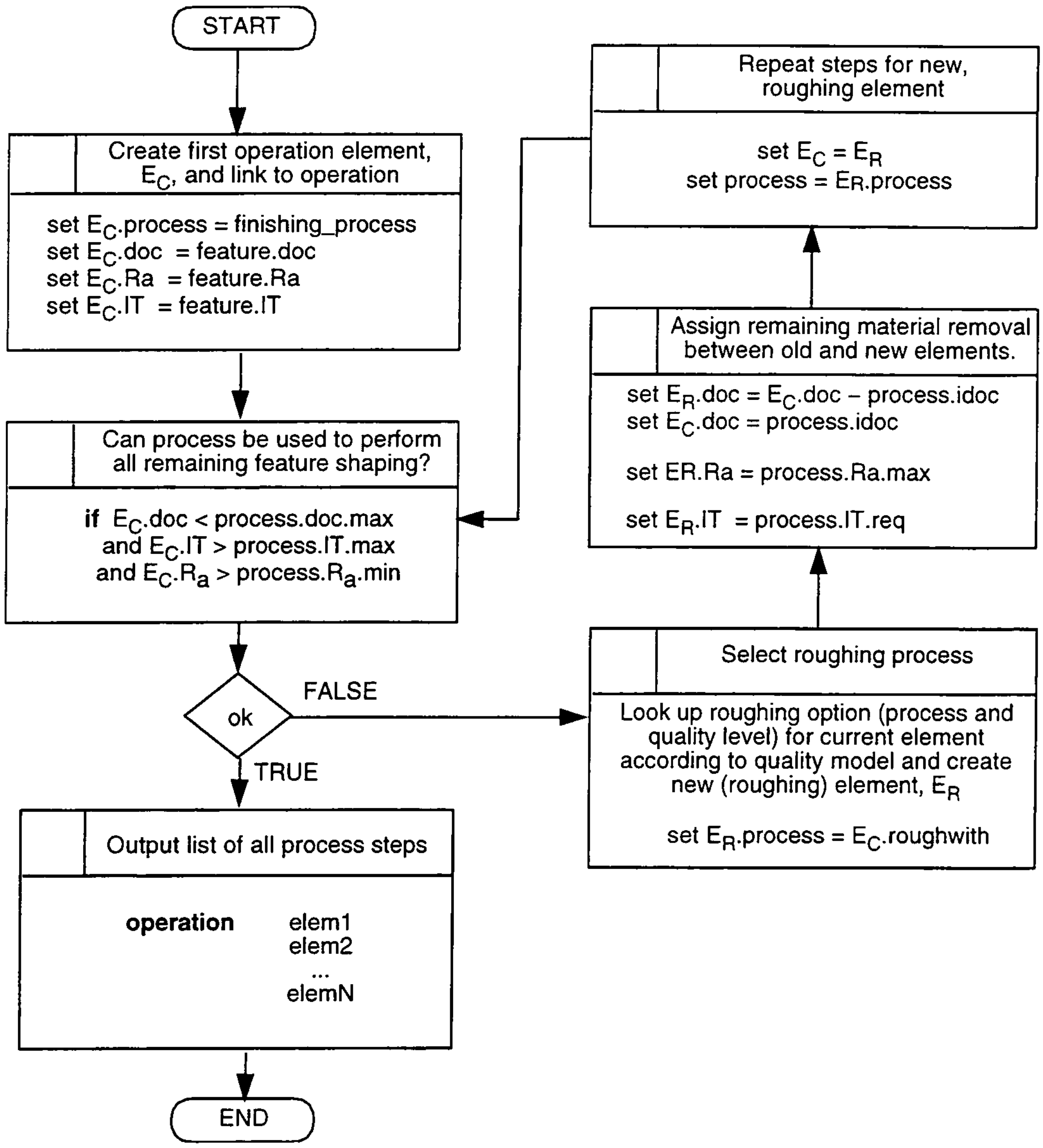


Figure 7.9: Creation of roughing elements

If the limits are exceeded, then the algorithm has to perform two tasks: firstly a suitable roughing process must be selected for a new operation element and secondly the depth of cut of the old operation element should be recalculated to account for a previous step. The roughing process to be used is defined for each process quality level. This may be another version of the same process (e.g. rough turning for semi-finish turning), or it may be a completely new process (e.g. drilling for rough boring). The amount of material removal to be assigned to each stage is determined by the element performed later: for each process quality level, an ideal depth of cut is defined which should be used when there will be previous roughing stages. This ensures that the maximum quality is achieved for the process and that the greatest proportion of the material removal will be done by the rougher process. The final stage of the algorithm is for the focus to be moved to the new operation element. The next iteration then begins, as the system checks whether this element can remove the remaining material. The end of the algorithm comes when all the initial depth of cut has been assigned.

7.5.4 Evaluating manufacturability indicators

In the fourth stage, once each operation has been fully detailed, CESS calculates the manufacturing times for each one, so that a selection of the most suitable processes for each feature can be made. At this stage the system makes use of the time calculation methods which are defined for each process in the machining process model, as detailed in Chapter 5. Each operation element inherits a particular method from its parent process class. A method takes as inputs the properties of the individual element and its parent feature, thus combining the product and process models in order to generate the route model. This structure is shown in Figure 7.10.

In addition to the process time calculation method, each element also inherits a machine tool type from the process class. This machine tool type refers to the classification of machine tools which is held within the system. Since a particular machine tool has not been selected at this stage, the system uses aggregate values for any machine related parameters which the process time methods may use. Examples of this include the machine tool power, as shown in the figure, and loading times, which are calculated using an algorithm based on empirical data relating component weights to each machine fixturing type. The aggregate machine tool parameters are calculated by taking the mean of that value for each example of that machine type available for planning consideration within the factory. Where the value is a calculated one such as the loading time mentioned above, then this value is calculated for each machine tool separately and the mean of these values is taken. This approach means that the times calculated assume that the machine tool which will be used is a "typical" machine tool. The time calculated will therefore be expected to be within the middle of the range of values calculated later for individual machines. The process quality indicator is not calculated for this stage of the aggregate process plan, since machine data required by the algorithm is not yet available.

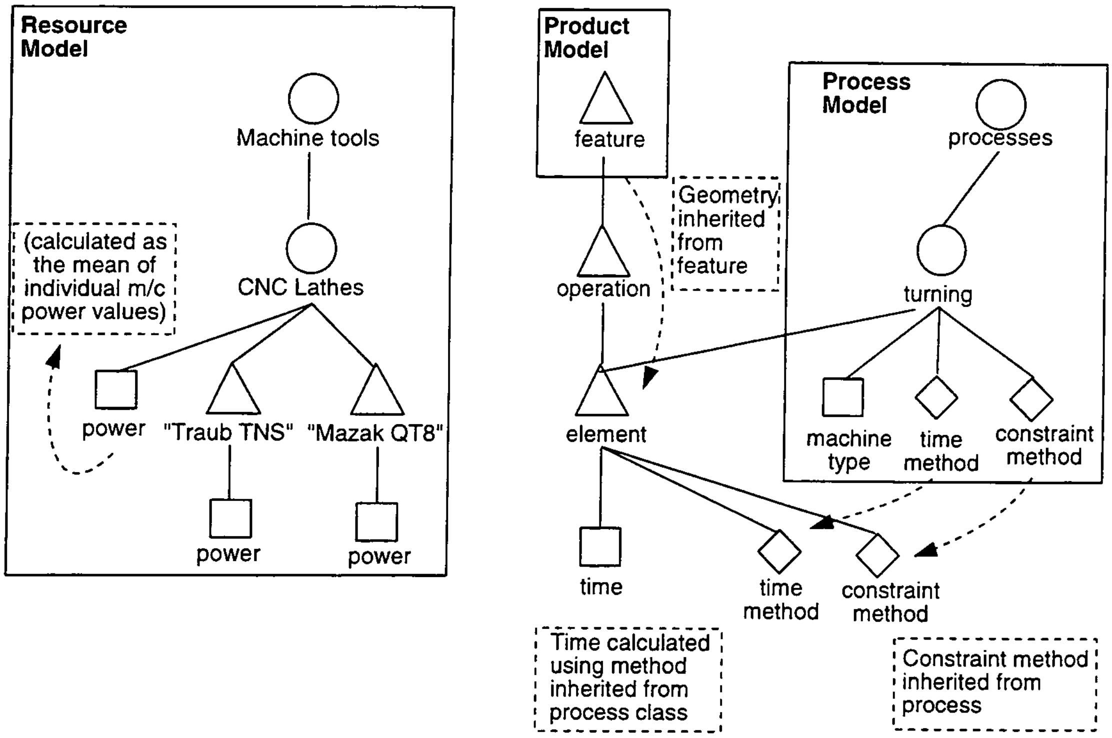


Figure 7.10: Operation element links to system models

7.6 Process selection functionality

This stage of the algorithm performs an optimised selection of the process alternatives for each feature to generate a number of alternative process sets which can be used to create the component. The optimisation is performed using a genetic algorithm technique based on an objective function of the overall production cost of the component. This total cost includes material, processing and transportation costs. As stated above, aggregate data representative of the set of available machines is used for the calculation of processing times, since machine tools have not been assigned at this stage.

A genetic algorithm uses a method analogous to biological evolution to generate better solutions from a *population*. A genetic algorithm modifies the solutions such as to minimise (or maximise) the *fitness*, calculated using the *objective function*. The population of alternative solutions are modified through a process of *breeding* and *mutation* so that the influence of each solution on the generation of future solutions is based on its relative fitness. At the end of the algorithm, the population will consist of a mixture of solutions with high fitness. In this example, a high fitness relates to a low overall cost. Figure 7.11 shows a schematic of the genetic algorithm concept.

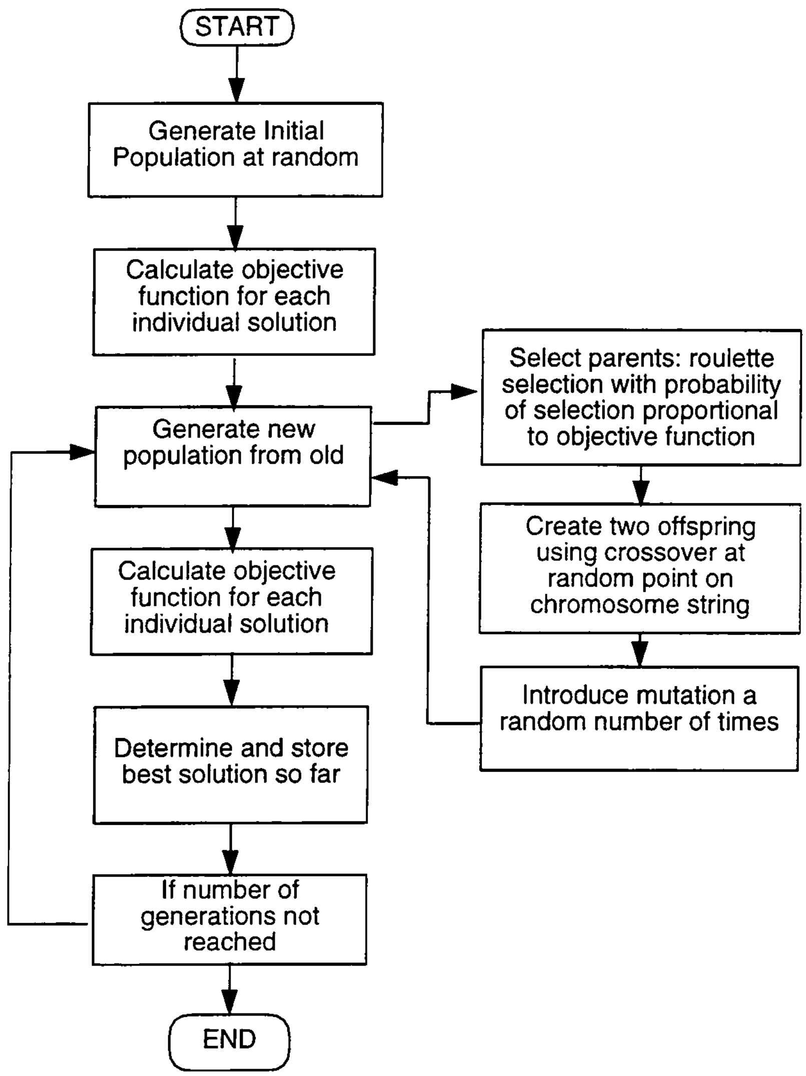


Figure 7.11: Process selection using a genetic algorithm

In choosing a suitable search algorithm for the optimisation, the chief criteria was to adopt an approach which would operate fully automatically in all cases. For some components, there will be a very small number of potential routes to be assessed, whereas for most components the number of potential routes will be large, in excess of 10^6. Another criterion influencing the choice of search technique was the need to generate alternative routes. Genetic algorithms are particularly suitable for this since the technique involves the maintenance of several alternative solutions at any one time, and this population should contain a set of the best routes at the end of the procedure.

In addition to genetic algorithms, other search techniques were considered: The extremely large number of potential solutions possible rules out a direct analysis of all possible combinations. Similarly, a purely random, or *shotgun*, search of the possible space would not provide a sufficiently efficient search method. More refined methods of the shotgun approach include the procedure of *shrinking* the search space by selecting a sub-region of the space through statistical evaluation of the results found. However, the drawback of this approach is that the global optimum may be missed, particularly if the search space is very irregular. The technique of simulated annealing has been used for process planning purposes. Simulated annealing algorithms are based on a simulation of annealing of metal, where the aim is to obtain a low energy state. In application, the cost of a process plan would be used as the energy of a solution. The algorithm processes by successively lowering the overall energy of system from a high value. Theoretically, at the end of the procedure the minimum energy will be reached if the cooling is infinitely slow. However, in real application, the cooling speed must be finite, which means that the chance of reaching the best solution is reduced. The slower the algorithm is run, the more likely the solution is to be the global optimum. Opinion is divided on whether simulated annealing algorithms are as efficient as genetic algorithms, since both require delicate control of the parameters. The genetic algorithm approach was preferred because it was simpler to implement in this instance.

In order to demonstrate the principle, a simple implementation of the genetic algorithm is used in CESS. This algorithm utilises a single population of fixed size, and the parameters of mutation and crossover are determined by global variables that are set by the user. The algorithm implements a moderate level of elitism to ensure that the best route identified in any one generation proceeds to the next. The algorithm works to a predetermined number of generations based on a calculation of the number of possible routes. Typically, the algorithm requires less than 100 generations to identify the near-optimum process route.

There is potential for further work to improve the selection system through adding sophistication to the genetic algorithm. In particular, it is possible to use the technique of niches to differentiate the search and thus propose less similar alternative routes at the end. Also, the algorithm could be modified to perform a multi-criteria search by changing the objective function and the use of spreading techniques.

7.6.1 Route encoding scheme

Potential production routes are represented during the genetic algorithm selection as a string of numbers expressed in binary notation. Each number represents the operation which has been selected from set of alternatives for that feature. The algorithm uses the minimum number of binary digits to represent the operation selected for each feature. Thus, if there are two alternatives, a single binary digit is sufficient, whilst if there are four alternatives, a two bit binary is required and for five to eight alternatives, a three bit binary must be used. The use of a binary notation gives the purest expression of the genetic algorithm method and means that the functions of mutation and crossover can be easily programmed. When the number of alternative operations for a feature is not a power of two, there will be the possibility of operation numbers higher than the actual options being chosen during the algorithm. Where this occurs, the whole individual is invalidated within the population by applying a penalty value to the objective function. This penalty function reduces the chance of the individual being used to generate the next population, so the bad operation choice should disappear from the population.

7.6.2 Objective function

Genetic algorithms rely on the determination of an appropriate objective function. This is the criteria by which alternative solutions are judged and the probability of their influencing the generation of further solutions is determined. The better the value of the objective function, the higher the probability that it will be selected for breeding and therefore that some part of it will be used in the generation of further solutions. For a minimisation problem, such as this case where cost is being minimised, the individuals in the population with the lowest value for the objective function will be assigned a higher probability to influence the next generation through crossover and breeding.

In the case of process selection, the objective function which is used is the overall production cost of the component. This overall cost is given by:

$$C = C_m + \sum_{elements}\left(R(t_m + t_p) + C_t\right) + \sum_{machines} Rt_b \quad [1]$$

where:

C	=	Total cost per unit (£)
C_m	=	Material cost per unit (£)
R	=	Machine cost rate (£/min)
t_m	=	Element processing time (min)
t_p	=	Element handling time per unit (min)
C_t	=	Transportation cost (£)
t_b	=	Machine set-up time per unit (min)

It can be seen that the total cost is made up of three components: material cost, element cost and machine cost. The element cost is the cost which is incurred for each element: this is a function of the actual processing time and the handling time required to set up the workpiece. At this stage, the algorithm assumes that each element requires the workpiece to be reset on the machine. Later in the algorithm, after machine selection when more information is available, the requirements for the number of set-ups are determined more accurately. In addition to element cost, there will be costs incurred due to the time required to set up each machine tool which is used. This is the machine cost factor, which is calculated by adding the average batch set-up time of each machine type which is used only once, no matter how many times it is specified in the plan for the component. This is a necessary approximation in the algorithm: at this stage, before individual machine tools have been selected and sequence determined, it is not feasible to determine how many individual machines will be required or how many set-ups will be required. The algorithm makes the assumption that only one machine tool of each type will be required and also that the machine tool will only need to be set-up once. This approach leads to an under-estimate of the required time and therefore cost at this stage, but since this is only an intermediary step, producing data which will be refined later, this is considered to be acceptable.

7.7 Machine Option Generation

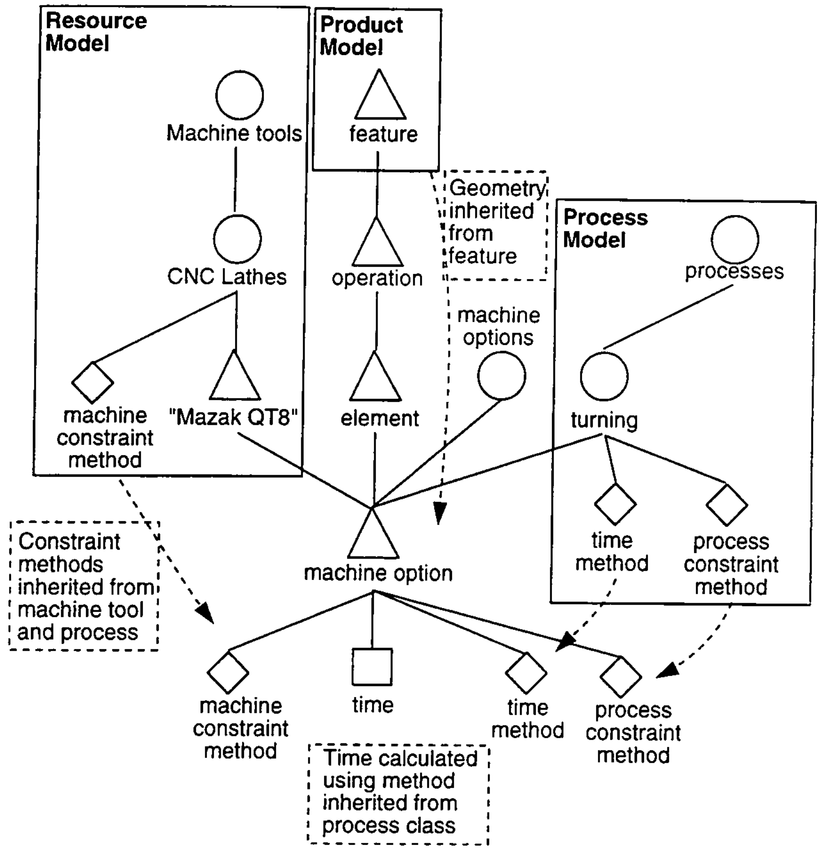


Figure 7.12: System model of machine options

This stage generates a list of machine tool options for each element of the processes which are selected for the route. Machines are drawn from the resource database which describes the factory to be used for manufacture. Constraints are applied for machine volumetric capacity and quality capability. As for process options, the machine options are evaluated according to the criteria of cost and quality. The costs are calculated using the same process time calculation methods which are used in the process selection stage. At this point, however, individual machines will been selected, and so the methods will use specific machine parameters.

Quality is modelled as a rate of scrap per operation, and is calculated from real factory data which is retrieved from historical machine SPC analysis. Historical quality performance is modelled statistically and is used to arrive at a prediction of expected quality for the current feature based on the tolerances specified. This method relies on the existence of a database of SPC results from real factory operations. The quality method performs a similarity search upon the SPC database, extracting similar operations which are ranked according to the degree of similarity and sample size. The method then retains a fixed sample size of the most highly ranked records, from which an average standard deviation is extracted, and used to predict the capability of the new operation. The system then applies Gauss's law to determine the probability of a part falling outside the allowed tolerance bands, which gives the expected scrap rate of the process. Since the APP is designed to operate on aggregate design information, there will be features for which the tolerances have not yet been specified. In this case, the system must select a standard tolerance interval (IT) and apply it to the feature so that the process selection methods can operate at an appropriate level and a process of sufficiently high quality can be chosen.

If no SPC data is available, then the system uses an alternative method of quality prediction. This method predicts the quality based on the size of the feature tolerance intervals and the limits defined for the process. The capability value for tolerances in between the upper and lower quality limits is interpolated, as described in section 5.6.5.

7.8 Sequencing

This stage is performed using a heuristic algorithm based on a knowledge based system which embodies production planning expertise. The system aims to satisfy the numerous constraints to determine an advantageous order to carry out the elements of the production plan. The constraints which must be satisfied are detailed below, after which the algorithm is described in detail.

7.8.1 Factors affecting sequence selection

The factors which affect sequence selection can be classified into three groups: Hard constraints, soft constraints and cost constraints. Hard constraints are those rules which must be followed because it is physically not possible to break them. These are typically related to geometry as explained below. Soft constraints are general user defined engineering rules which should be followed in order to improve the product quality or simplify production. These are rules which can be broken *in extremis*, for example if they conflict with the hard constraints or they cause very large costs. Cost constraints are general rules which are aimed at reducing costs. These are applied when they do not conflict with the other rules.

Hard Constraints

Geometric	Geometrical precedence relationships between features, where a particular feature must be created before a second can be made, are common. A simple example is an internal thread, which requires the related hole feature to be created first.
Process grade	For a given feature, the roughing elements must be performed before the finishing elements.
Quality	Where features are connected by a quality feature relation, the datum feature should be created before the referring feature.
Access	Internal features that can only be performed after the primary internal material removal operation.

Soft Constraints

Process technology	Certain processes create constraints. In particular, when turning is used all axi-symmetric features should be created before any asymmetric features, since these will create an eccentric centre of gravity and therefore an out of balance load and vibration,

which reduces process quality.

Process precedence	Processes can be ordered into a general sequence which they should follow. In particular, processes which are generally used for large scale material removal should be performed before finer processes suitable for finishing operations. All heavy machining should be completed before finishing to avoid deforming the finished features.
User defined constraints	The user of the system may wish to intervene in the sequence selection process to specify additional constraints where the CESS model would not account for known special cases. The user may wish to cluster certain elements within the same set-up, or to specify that two or more separate set-ups are required because of a given geometry.

Cost Factors

Process clustering	Clustering the operation elements according to the process used for manufacture allows the machine selection algorithm to benefit from specifying the same machine for multiple adjacent elements.
Quality grade clustering	Clustering the elements according to process grade will potentially reduce the number of tool changes required and save on processing down time and cost.

It can be seen that certain of these constraints will cause conflict in the generation of a suitable sequence (e.g. process and quality grade clustering, which always conflict). A process plan will always require that compromises are made in order to arrive at a working solution.

7.8.2 Sequencing algorithm

The sequencing algorithm, shown in Figure 7.13, operates in a four of stages: First the operation elements are assigned weightings according to feature, process and process quality level. The second stage is to order the elements according to these weightings, yielding a provisional sequence. The third stage applies the soft constraints, whilst the fourth applies the hard constraints which can override the previous constraints.

7.8.2.1 Sequence weightings

In the first stage, a sequence priority index which is assigned to each operation element. This index is based on *weightings* which are determined by a set of rules which interrogate aspects of the element corresponding to process type, feature type and quality level. The weighting set by each rule is summed to give the priority index of the operation element, establishing a provisional sequence. For each sequencing criterion the variation in priority index differs by an order of magnitude, so that one criterion takes precedence over another. The order of sequencing is by process type first, then by quality level and then by feature type. These weightings have been set according to the cost factors identified above.

(1) Process class sequence index (S_p)

The first value to be added to the sequence index is the process class. This will define the initial sequence of the elements based on the process. General precedences for each process have been determined in order to generate the table of process indices. The absolute value of a process index is not important, merely the sequence. Note that turning and boring are given equal priority. The feature type will determine the sequence of these elements.

Table 7.1: Process Sequence Index

Sawing	2000
Turning	1400
Boring	1400
Planing	1200
Broaching	1000
Milling	800
Drilling	600
Grinding	0

(2) Feature class sequence index (S_f)

Table 7.2: Feature Sequence Index

internal cylinder	icy	100	prismatic face	pfa	20
internal step	isp	95	square shoulder	psd	15
internal profile	ipf	90	chamfer	pcf	10
internal groove	igv	85	prismatic thread	ptd	5
internal taper	itp	80	slot	pst	0
internal thread	itd	75	closed slot	cst	-5
external step	esp	70	v-slot	vst	-10
external cylinder	ecy	65	keyway	pky	-15
external profile	epf	60	pocket	ppk	-20
external groove	egv	55	prismatic through hole	pho	-25
external taper	etp	50	prismatic blind hole	bho	-30
external ring	erg	45	prismatic hole groove	pgv	-35
external thread	etd	40	primatic hole counterbore	pcb	-40
external face	efa	35	prismatic hole countersink	pcs	-45
2-D profile	sf2	30	prismatic hole chamfer	pcf	-50
3-D surface	sf3	25	thread	htd	-55

For each element, the sequence index is increased by the value of the feature sequence index value defined for the feature to which it is related. The values of the feature sequence indices are defined such that the feature value is subordinate to the process level, i.e., the largest feature index is less than the difference between any two processes. As shown, the order of sequence for the features runs from axi-symmetric to prismatic. Of the axi-symmetric features, the internal features are scheduled before the external. In general, features which tend to be large, such as shoulders and slots are given priority over minor features such as chamfers and threads.

(4) Quality grade sequence index (S_q)

Where a process has a number of different quality grades, the finishing elements are separated from the roughing elements through the addition of the quality grade sequence value to the sequence index. This value is defined such that the quality grade is subordinate to the feature selection in sequence determination, since the interval between feature indices is larger than the greatest quality index. The value is defined according to the following table:

Table 7.3: Quality grade sequence index

rough processing	4
semi-finish processing	3
finish processing	2
specialist finishing	1

7.8.2.2 Provisional sequence

Once a sequence index has been assigned to each operation element, the next step of the algorithm is to order the operation elements according to this sequence. This sorting will potentially separate elements belonging to the same operation in order to allow the grouping of similar processes. For example, if two features both require drilling before reaming, then the drilling operation elements will both be carried out before the two reaming operation elements.

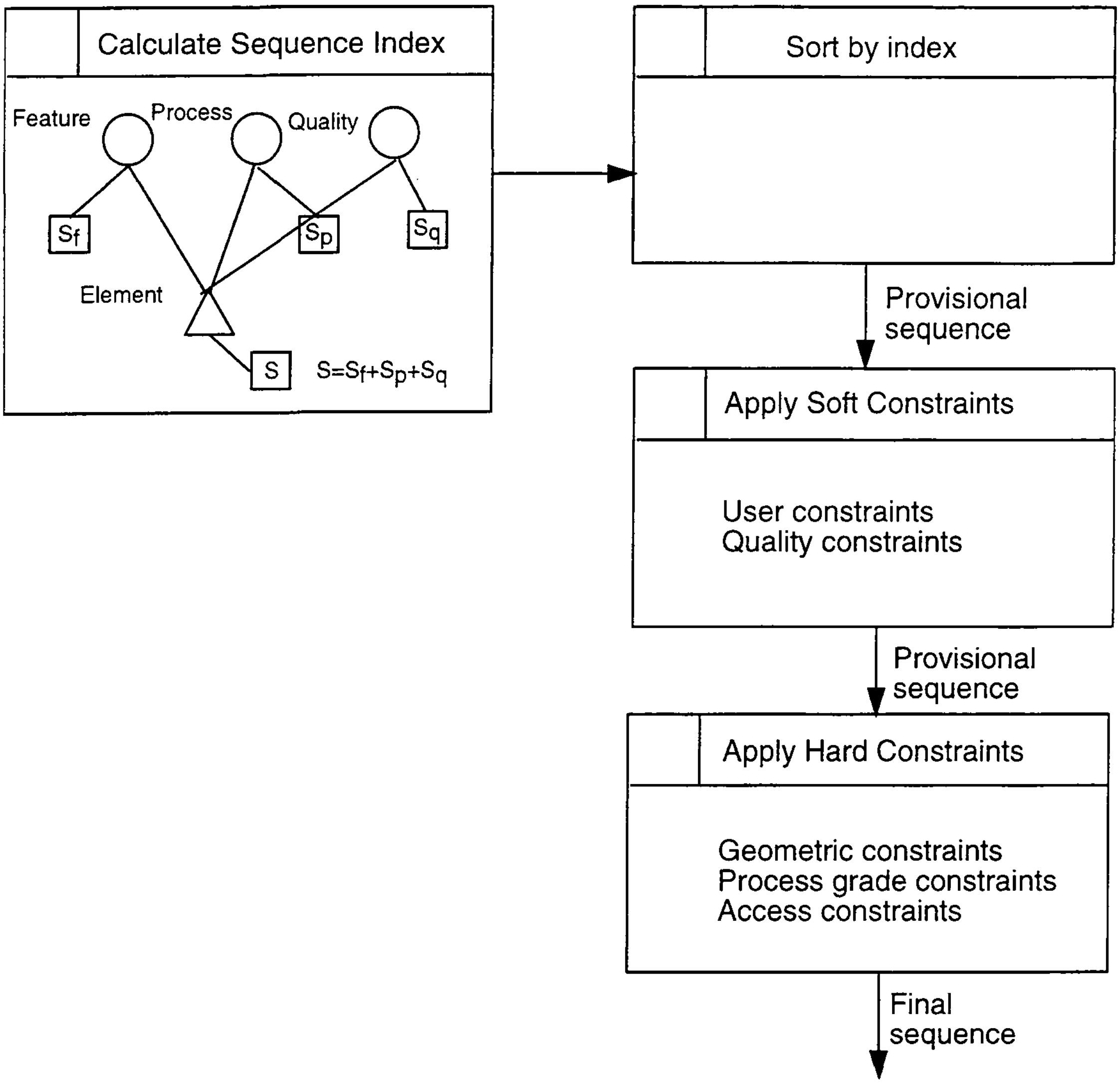


Figure 7.13: Sequence generation algorithm

7.8.2.3 Soft constraints

This stage applies the *soft constraints* such as user clustering of features. These are user specified preferences and which can be overridden by the *hard constraints* which are applied in the fourth stage. At this stage, a provisional sequence has been established for the elements. The next step is to apply the specific constraints, by stepping through the provisional sequence and testing each constraint. If a constraint violation is discovered, the system will rectify this by moving the element which must be performed first ahead of the current element in the sequence. The sequence test then moves back in the sequence to consider the newly positioned element, applying the constraints once again.

Once the generic constraints have been satisfied, the system checks through the sequence to find any user constraints. Where the user has specified that two features are within the same set-up, the finishing elements of these features are clustered together. Elements are moved to the position of the lowest ranked of those within the cluster, whilst retaining their relative positions to each other. Since user constraints are applied before the specific constraints of access and quality, these latter factors may override the user's request and separate the elements.

7.8.2.4 Hard constraints

The final stage of the algorithm acts to satisfy the hard constraints such as roughing before finishing on individual features, and datum features before their related features. For access constraints, the system checks the current element against all future elements. If this element has a requirement for one of the later elements to be performed first, it is moved as described for the soft constraints. The system checks for any quality datum links to later elements in the sequence. If any are present, the datum element or elements are moved ahead of the current element. Datum links are only applied to finishing elements.

When the system has applied all the constraints, the sequence is submitted to the user for scrutiny before the process planning system is allowed to continue. The user is then permitted to make any adjustments which are desired. This approach gives the user the ultimate control over sequence and allows the resolution of any conflicts between constraints.

7.9 Machines selection functionality

The final stage of route generation is to assign factory resources to the operations which have been identified. The most critical resource is the machine tool or work station. A factory may have a number of machines which are capable of performing a given operation. The task of the machine selection algorithm is to determine the best machine to perform each part of the operations within the process plan. This decision should take into account the cost, quality and lead time of the product. The machine selection stage uses a second genetic algorithm which operates on similar lines to the first. The objective function is once again the overall component cost. At this stage, additional planning has been carried out and individual machines chosen. Therefore, the calculated times and costs will be in relation to specific machines and be will more accurate. In addition, the system has calculated expected quality levels which are expressed as a scrap rate. This enables the objective function to incorporate the cost of quality for the component. The machine selection algorithm is detailed below. Once the genetic algorithm has been run, the output population can be browsed by the user in order to select the most suitable solutions. This may be the solution with the lowest cost, or a combination of fairly low cost with high quality and short lead time. The user interface allows the population of solutions to be plotted as a graph of cost against quality or lead time.

7.9.1 Objective Function

At this stage in the route planning algorithm, there is more detail about the process plan, and therefore the objective function of the second genetic algorithm is more complex than the first. The system makes use of all available information to calculate the predicted costs and qualities as accurately as possible. The objective function is based on the cost of producing the part. There are several contributing factors to the overall cost, which include:

- Material cost

  The material cost of the part is determined from the cost of the blank workpiece. In CESS the cost is calculated from a standard cost per weight for the material type, unless a specific value is given by the user. This latter approach would be used where the blank is a partially manufactured component.

- Machine tool cost

The second major cost factor is the cost of using the machine tool. This cost is proportional to the time which the component spends on each machine. Machine tool use cost can be divided into value-adding time, when the processing is taking place, and non-value-adding time, where the machine is being set up, and the parts are being loaded and unloaded. In CESS, three separate times are defined for the machine tool time:

t_m Processing time (actual machining time)

t_b Batch set-up time (time taken to ready the machine for a new batch, general changeover time)

t_p Piece handling time (time taken to load/unload the individual workpiece)

Some machine tools have off-line loading systems which can effectively reduce piece handling time to a negligible amount. The times are converted into costs using a cost rate which is calculated for each machine based on its purchase cost, depreciation and running costs.

- Labour cost

The cost of labour for a given process varies in importance with the process type. For highly automated machining the cost is negligible when compared to the machine costs, whereas for labour intensive processes like fabrication, the labour cost may be the most important cost. The cost of labour is a function of the time required and the rate of pay.

- Quality cost

The quality cost is a representation of the production quality which is achieved and of the product quality. The quality cost includes the cost of excess production required to cover for scrap rates.

- Overhead costs

  The overheads of a manufacturing company must be paid for through the profits on products, and therefore it is necessary to assign overhead costs individually to each product to reflect the true cost of manufacture to the company.

- Investment cost

  The investment cost represents the time value of the money which is tied up in the production of the process. This is, therefore, an indication of the cost of the factory inventory, and thus the cost consequences of a given production lead time. The shorter the production lead time, the sooner the customer receives the product and therefore the sooner the company recoups the investment through payment. In many companies this cost is overlooked, but where lead times are large then this cost can be significant. It is through the consideration of this cost that the trade-off between shorter lead times and higher processing costs can be investigated.

The objective function of the genetic algorithm must sum all these costs together for the component and return a single cost per component which allows all the different route options to be compared. When using a genetic algorithm it is important to minimise the computational load within the iterations so that the processing speed is increased. Where possible, therefore, the individual costs at operation element level are calculated outside of the genetic algorithm.

Once the elements which make up the route have been determined, the objective function calculates the total component production costs as follows:

1. Set running total cost equal to material cost.

2. Sum the individual processing costs for each element ($\Sigma\ C_p$).

3. For each element in sequence, add the transfer cost from the previous machine ($C_t[m_{i-1}\text{->}m_i]$).

4. For each element in sequence, add the piece handling cost if the machine or set-up number has changed (C_h where m_i=m_{i-1} OR $s_i \neq s_{i-1}$).

5. For each element, calculate the quality cost as the scrap rate times the running total cost and add to the running total cost.

6. Add the machine set-up cost for each machine visited ($\Sigma\ C_b/b$).

7. Once all elements are costed, multiply total cost by the overhead factor, if any.

Where: C_p = Machining cost per unit (£),

C_b = Machine set-up cost (£),

C_t = Transfer cost per unit (£),

b = Batch size

m_i = Current machine tool index,

m_{i-1} = Previous machine tool index,

C_h = Piece set-up cost per unit (£),

s_i = Set-up number of current element.

It can be seen that the objective function of the second genetic algorithm is complex. However, this is necessary to determine the costs accurately. This cost algorithm correctly assigns higher costs where the workpiece is moved around amongst many machines since the cost of loading and moving the workpiece is calculated each time it moves. In addition, it can be seen that since quality costs are calculated sequentially through the route, the benefits of reducing scrap rates increase as the workpiece contains more added value. If a workpiece is scrapped after only one operation, less effort is wasted than if it is scrapped after nine.

7.10 Conclusions

Generic models have been developed to represent the product design, process knowledge, production resources and production route plans, using an object oriented system. These models have been integrated into a system to allow the automated generation of aggregate process plans from aggregate design data, and the calculation of costs and quality for these plans.

An automated process planning system has been developed which provides an assessment of the costing and quality for the design. The process plans are produced generatively from a set of basic production planning rules store in the process model using the aggregate product model as the input. The routing module generates a number of alternative options for processes and then for machines, and selects the most suitable options based on cost using a two stage genetic algorithm technique. This approach allows the user provide an input into the system at various stages to influence the route generation where preferences exist. On the other hand, the system is capable of producing a route without reference to the user. A key strength of the genetic algorithm approach is that alternative routings are generated as a matter of course, and these may be retrieved and supplied to the user for consideration.

Chapter 8

Testing and results

8.1 Introduction

With a research project such as this, the design of an effective method of testing and evaluating the concepts and theories proposed gives rise to difficulties not found with a more experimental work. In particular, it should be recognised that the most informative method of testing the work is not feasible: For a manufacturing company to adopt an untried CAE tool as a central part of its product development strategy would be to take unacceptable risks. It is therefore necessary to look for alternative ways of evaluating the methodology and software developed.

The principal evaluation strategy which was adopted in this project was to analyse the performance of the computer system when performing product development tasks on example product designs. The system outputs were then compared with the information available from traditional methods and industrial data. Data gathering from industry proved to be a problem, since companies frequently do not have process cost breakdowns at a feature level. Differences in costing methods also made it difficult to find comparable data figures to that produced by CESS.

Two methods were used for the testing of the CESS models and functionality and these are described below:

1. Build example component models within the system.

Typical component designs can be extracted from a product and modelled in the CESS product model. This enables the demonstration of the product model flexibility, the range of geometries to which it may be applied and the concept of aggregate product modelling. Two approaches may be taken: modelling of selected components to emphasise particular modelling functions and modelling of randomly selected components to test the generic nature of the model. In this project, the model was initially developed using components from Warner Electric Ltd. Additional testing was performed using products from other sources.

2. Generate aggregate process plans for example designs using the system.

The aggregate process planning functionality can be demonstrated and tested through the use of the example product models. It is necessary to demonstrate a number of aspects of aggregate process planning:

(i) It can be applied to a variety of product model configurations,

(ii) It can produce alternative production options for the same design, identifying alternative processes automatically,

(iii) Suitable processes are always evaluated,

(iv) Suitable machines are evaluated and the proposed routings are realistic,

(v) The process plans produced are both technically feasible and realistic (i.e. no machine or process constraints are violated),

(vi) Estimated times and quality levels calculated are sufficiently accurate,

(vii) Process plans are produced in a sufficiently automated way in an acceptable time scale (CESS is intended for use as a rapid evaluation tool, so that it may be run many times as the design continually evolves).

Many of these criteria relate to the overall functioning of the system. There are others which relate more specifically to individual modules within the system. The tests which have been carried out were designed to assess the criteria by running only the necessary parts of the system when possible, in order to reduce the time required for testing. The aggregate process planning evaluation can be divided into stages according to the main planning steps: process option identification, process evaluation, process selection, machine option identification, sequencing, machine option evaluation and final selection.

8.2 Product model tests

The product model was initially developed and tested by representing components which form parts of a clutch assembly, designed and manufactured by Warner Electric (UK) ltd. Further proving of the product model was then achieved by using the system to model components forming part of a larger structural assembly. These products were chosen for the presence of both axi-symmetric and prismatic features.

8.2.1 *Moving armature component*

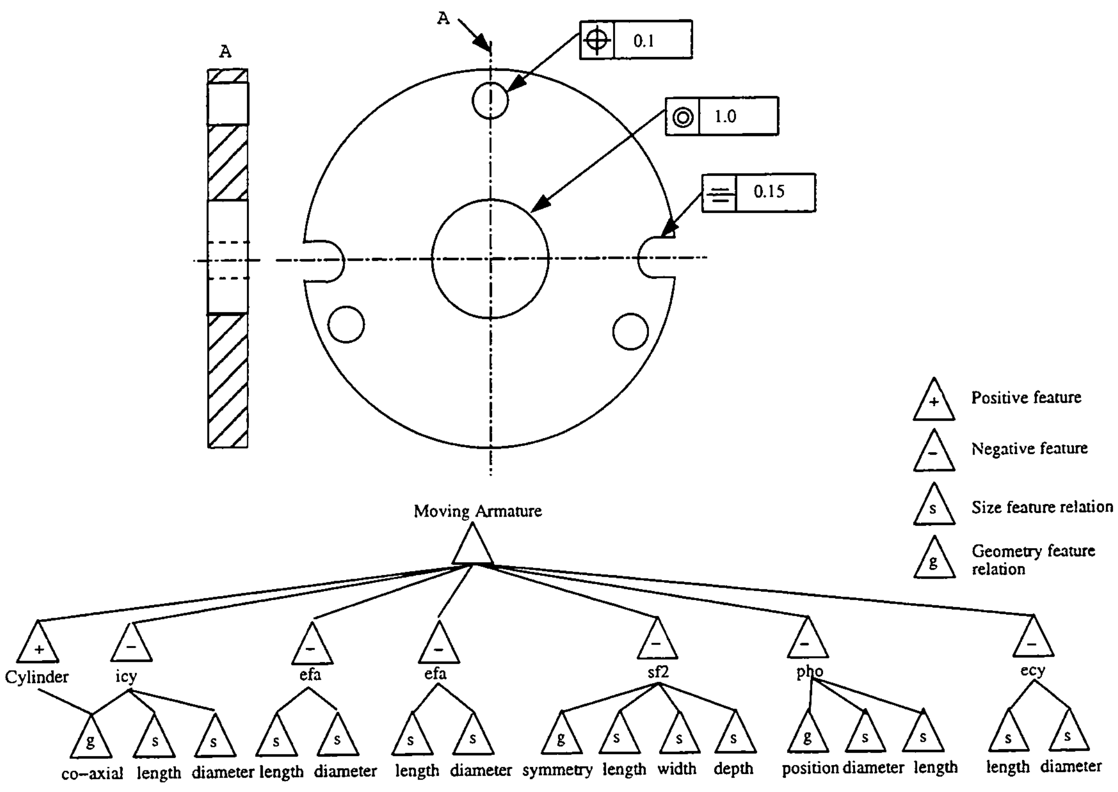


Figure 8.1: The product model for the armature component

Figure 8.1 shows the product model for the moving armature component. The component has been modelled using a single positive feature (the cylinder), and six negative features. Each of the negative features has two or more feature relations defining the geometry. The overall feature component dimensions are a diameter of 100mm and a length of 6mm.

To create an aggregate product model, the selection of the positive feature is critical. It is possible that there will be more than one positive feature which could represent the basic component shape. In these circumstances, the selection of the best feature should be made on the basis of preserving design intent. It should be recognised that the shape of the positive feature will influence the possible processing options. The armature component is a relatively flat component and therefore the designer might have chosen to base the product on a sheet feature. In this case, however, the processing options available would have excluded axi-symmetric processes such as turning and boring. The cylindrical representation is more flexible, since it allows the consideration of turning processes as well as prismatic processes such as milling.

Since all surfaces of the component must be finished to a high degree, features have been specified for the end faces (two *efa* features, see section 4.4.4 for descriptions of feature types) and for the edge (*ecy*). The efa features present an example of the use of the set-up attribute of the feature class: The two end faces cannot be machined within the same set-up on a lathe, since when the workpiece is held in a chuck, only one face is accessible. This means that two separate features must be specified, one for each face. The process planning system is forced to place them in different set-ups by assigning different values to the set-up attribute. In this example, two separate features would be required for the ends in any case, since they have different surface finish requirements. In contrast to the faces, a single feature (*pho*) has been used to represent the three small holes through the component. This has the effect of forcing the same processing method and set-up to be selected by the process planner and at the same time reducing the computational load. The same feature object can be used for all the holes because the features are identical except for their location. The remaining features used to define the component are an internal cylinder (*icy*) which defines the axi-symmetric hole and a two-dimensional surface (*sf2*) feature which represents the two slots which lie opposite each other. For the latter feature, there was a choice to make between a pocket feature, a slot feature and the surface. The selection of the surface feature was made because it is the simplest method of approximating the geometry. Using a slot feature would have required additional feature relations to model the rounds, whilst the pocket was unnecessary since the slots are "through", with easy access. The full product model description for the moving armature which was used to produce the results shown later in this chapter is shown in Appendix E, Table 1.

The moving armature component has three geometrical tolerance specifications, which are shown on the component drawing. Each of these tolerances is represented in the product model tree by a feature relation. These feature relations can have two parent features, as is the case with the coaxial feature relation which is attached to the *icy* and *cylinder* features. When this is the case, one feature is always identified as the datum feature using a property of the feature relation. In this example it is the *cylinder* feature. The position and symmetry feature relations are connected to just a single feature each.

8.2.2 Flange component

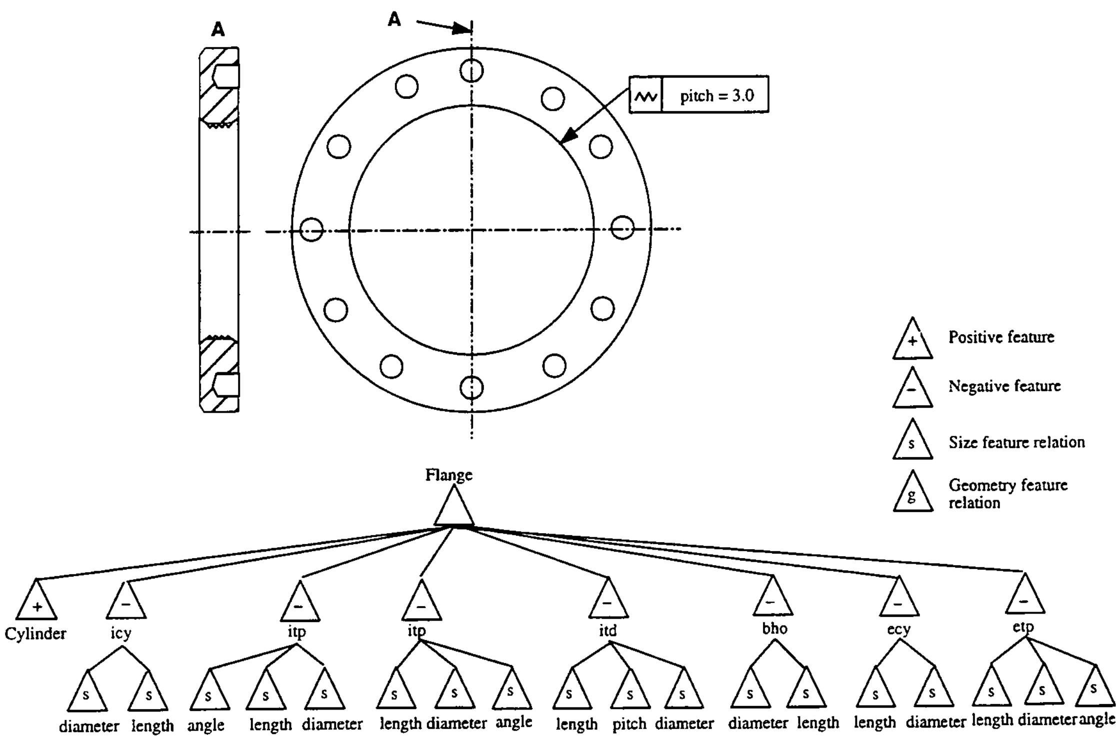


Figure 8.2: Flange component: product model

The flange component is a large, hollow cylindrical component, much wider than it is long, which has several large diameter axi-symmetric features that will dictate the use of certain processing methods. The overall size of the component is 407mm external diameter by 32mm length. The component is represented by a single positive feature, a cylinder, and by seven negative features, as shown in Figure 8.2. Of these negative features, six are axi-symmetric, whilst only one is prismatic. The axi-symmetric features include: An external cylinder feature (*ecy*) which is used to specify the diameter and surface finish of the cylinder; chamfer features where the faces meet the external (*etp*) and internal (*itp*) cylinders; a large diameter internal cylinder (*icy*) and an internal thread on this bore. The thread, is of particular interest, since with a diameter of 295mm it is too large for many threading processes. The prismatic feature is the blind hole feature (*bho*) which represents the twelve radially spaced holes. The presence of this prismatic feature will prevent the machining of the whole component on a lathe. The full product model description for the flange is set out in Appendix E, Table 2.

When developing process plans for this component, the manufacture of the internal cylinder will have an important impact on the creation of the other internal features. Each of these internal features will require that the internal diameter is generated before they can be made. The quality to which the internal bore is made will affect the production of subsequent features, particularly the thread. No quality feature relations have been specified in this example. The final detailed design would probably include such requirements, particularly for geometrical tolerances on the internal cylinder, which would have concentricity and circularity requirements as the datum for a thread. The blind holes might also require a positional tolerance. In this case, however, the product model represents the design during the embodiment stage, when features are being added but not all constraints have been determined.

8.3 Factory model used in testing

In order to run CESS for testing purposes, a factory resource model was required. The machine tools from Warner Electric were modelled according to the methods described in Chapter 6 and used for this purpose. Figure 8.3 shows the factory layout window from CESS, which shows the positions of the machines within the factory. The factory is divided into seven cells, as shown. Of these, the coils cell is dedicated to the production of solenoid coil components, therefore it contains no machining equipment. The machine tools in the friction cell are equipped with extraction equipment and so this cell is dedicated for machining components made from hazardous material. These machines are not therefore available for general use. The press shop is shown with representative examples of the presses which it contains. These presses are used only for sheet metal work and are not modelled in CESS at present. The jobbing cell contains manual machine tools which are only used for low volume work. This cell is primarily used to manufacture prototypes. The drill cell contains specialised drilling and tapping equipment. These machines are manually operated. The two main cells of the factory are the ERD and ERS cells, which are named after the product ranges for which they are used. These machine tools are generally CNC lathes and machining centres. The machine tools which have been labelled in the figure are those which have been mentioned later in this chapter. A full list of the factory model is shown in Appendix F.

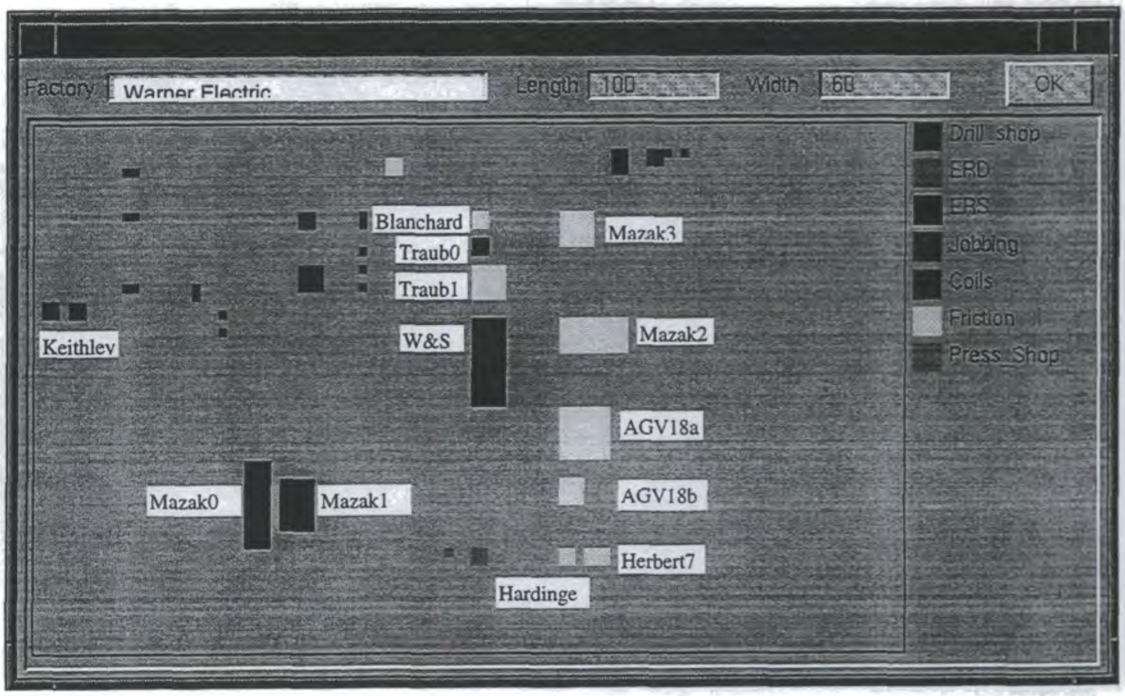


Figure 8.3: Warner Electric factory layout

The machine tools labelled belong to the following types:

CNC Lathes: Mazak0, Mazak1, Mazak2, Mazak3, Traub1, Traub2, Hardinge, Herbert7.

Machining centres: W&S (Warner and Swasey), Mazak AGV18(a) and (b).

Grinding machines: Keithley, Blanchard

Some of the above machine tools are identical models, such as the pairs Mazak0 and 1 and Mazak2 and 3. In these cases, when the test examples were run, the machine selection algorithm was instructed to ignore one of the machine tools. Machine tools can be removed from consideration using the factory layout window. This would be useful when machines were dedicated to other products, or when a machine was unavailable due to maintenance requirements or high capacity utilisation. It is also possible to restrict planning to a cell or selection of cells. This might be desirable when a cell is being designed for a single product range. In the examples shown below, however, the aggregate process planner was left free to select from machines in any cell in order to enhance variety and make the selection process more difficult, and therefore a more useful test.

8.4 Testing process planning functionality

The testing of the aggregate process planning will be described in two sections. Firstly, in this section, simple examples will be used to demonstrate the functionality. Three case studies will explore the planning at the level of individual features, showing the operation of the process generation, machine selection and operation evaluation routines. Two further examples will present the planning for more complex components having multiple features. In the next section, aggregate process planning for the real industrial components described previously in this chapter will be presented.

The aggregate process planning output presented in this thesis has been formatted into tables from data files saved by CESS during the example runs. For each planning run, the data is divided into two tables. The first of these tables is taken from after the process selection stage of the algorithm. At this stage, four aggregate process planning stages have been performed: process identification, process detailing and process evaluation. In the first stage the alternative process options for each feature are identified. These options are checked to remove those which have constraints that are violated. In the second stage the process options are been detailed and divided into appropriate roughing and finishing steps where required. The third stage is the calculation of processing and set-up times for each of these steps according to the relevant process model. Finally, the system generates a number of alternative process selections for the component using the first of the two genetic algorithm modules. The data presented in the first table is therefore a breakdown of each of the alternative process options for each feature. An example table is presented in Table 8.1. Each row represents an *operation element*, i.e. a single processing step. The "feature" column identifies the feature which the step is producing. The "op." column identifies which alternative *operation* the step is related to, since each operation element within the same operation has the same number. The "elem." column is an index for the processing step. Thus, the example table shows that the external step feature has three alternative operations, i.e. three alternative ways of being made. The external face feature has only two operations, the second of which has two elements representing two stages of processing. For each operation element, the table shows the process which is used, the required machine type, the piece set-up time (*Tsetup*) and the machining time (*Tmachining*) in minutes. Also, the table shows the best of the alternative combinations of process options which was found in the process selection search. The operations which are selected are shown as shaded lines on the table.

Table 8.1: Example of process alternatives output

feature	op.	elem.	Tsetup	Tmachining	machine type	process
external step	0	0	0.5	0.7	grinding_mc	cyl. grinding
"	1	0	0.5	0.4	lathes	profile turning
"	2	0	0.5	0.2	lathes	rough cyl. turning
external face	0	0	0.5	0.8	grinding_mc	face grinding
"	1	0	0.5	0.2	lathes	finish facing
"	1	1	0.5	0.4	lathes	rough facing

The second table shows the final results of the aggregate process planning stage. This incorporates the results of the machine option identification, machine option evaluation, sequencing and machine selection stages. Since multiple alternative routes are produced by the selection algorithm, a number of alternative routes may be shown. The table may show several such routes, each labelled with a letter, each a full plan for the component. An example of this table in shown in Table 8.2. The rows on this table again correspond to the operation elements. At this stage, the elements have been sequenced for production and therefore operation elements for the same feature may no longer be adjacent. The feature column identifies the relevant feature. Each row is numbered, representing the sequence, starting from zero for the first production step. The table shows the material cost of the component (£) and the following data for each operation element:

machine The name of the machine tool allocated to the operation element

tm Machining time. The actual time spent machining the workpiece (minutes).

tp Piece set-up time. The time in minutes required to set the workpiece up on the tool. This is only added when the plan calls for a new set-up on the same machine or for a change in machine tool (minutes).

tt The time required to transfer from one machine tool to another (minutes). This is only added when the workpiece is moved between machine tools.

cost The unit production cost for each operation element, including set-up costs, processing costs, transfer costs and quality costs (£).

sum The cumulative total cost of production for the component (£). The final value of this cost is the total predicted cost of the component.

Table 8.2: Example aggregate process plan output

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
A								
	Material						2.50	2.50
0	external step	rough cyl. turning	m/c 1	0.19	0.45	0.00	1.28	3.78
1	external face	rough facing	m/c 1	0.38	0.00	0.00	0.79	4.57
2	external face	finish facing	m/c 1	0.19	0.00	0.00	0.38	4.95
B								
	Material						2.50	2.50
0	external step	rough cyl. turning	m/c 2	0.21	0.55	0.00	1.52	4.02
1	external face	rough facing	m/c 2	0.42	0.00	0.00	0.84	4.86
2	external face	finish facing	m/c 1	0.19	0.45	0.10	1.28	6.14

8.4.1 Example 1: An external cylindrical feature

In this example, a component has been created with a single feature. An axi-symmetric component with a single external step feature has been defined. This example will be used to demonstrate the effect of varying the quality limits specified for the component feature. The product model of the component is shown in Figure 8.4, together with a sketch of the finished component. The product model diagram shows the classes of which the various product model objects are instances. The component, named *Test Component*, is an instance of both the *products* and *components* classes. The overall component shape was defined using a cylinder positive feature of length 100mm and diameter 50mm. The cylinder object is a member of the *cylinder* class, which is a sub-class of *positive features*. The external step is an instance of the *esp* sub-class of *negative features*, whilst the dimensions of the feature are specified as feature relation objects belonging to the same *size* sub-class of the *feature relations* class.

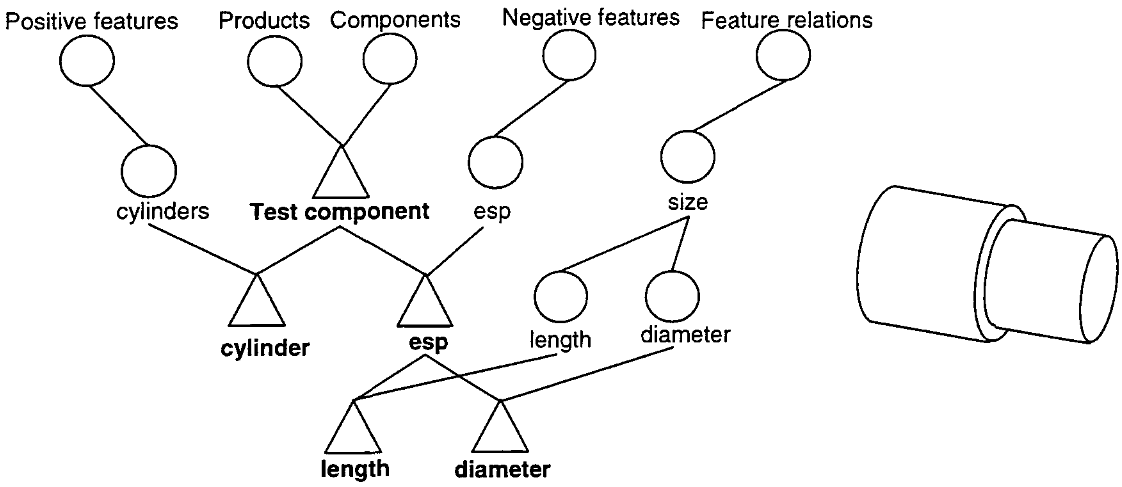


Figure 8.4: Simple component with one negative feature

Table 8.3 gives a detailed breakdown of properties of the *esp* feature object and the feature relation objects. The external step feature has a diameter of 40mm and a length of 50mm, half the component length. The diameter implies a reduction of 10mm in the component diameter, requiring a radial cut of 5mm depth. The tolerances as shown are set to an interval of 1mm, the equivalent of a standard tolerance grade of IT=15. The component material was defined as mild steel. This component will be run with varying tolerances to demonstrate the aggregate process planning method's use of roughing and finishing stages and the increase in cost which is caused by higher quality requirements.

Table 8.3: External axi-symmetric step product model

Name	PARENTS	PROPERTIES	VALUES
esp872162220	esp Test_Component	component number rank selfname set-up typeclass prefix it roughness	Test_Component 1 Unknown esp872162220 1 features esp Unknown Unknown
length_esp872162220	length esp872162220	upper lower nominal typeclass	0.5 0.5 50.0 geometry
diameter_esp872162220	diameter esp872162220	upper lower nominal typeclass	0.5 0.5 40.0 geometry

(a) The results of aggregate process planning for IT=15

The results of the first phase of aggregate process planning show all the alternative processes which have been identified for finishing this feature to the specified tolerance. Table 8.4 shows that the system identified four alternative process for finishing the component. Of these, three are variants of turning, whilst the other is cylindrical grinding. Each of the three turning variants (cylindrical turning, profile turning and facing) requires only a single stage of processing, corresponding to one operation element. The turning variants reflect alternative modes of using the generic turning process. Cylindrical turning cuts with the tool moving parallel to the axis of cutting, whilst in facing it moves radially. In profile turning the tool may move in both directions at once, although in this instance a single direction would be selected. The relatively great length, compared to the depth, of this feature means that facing would require many short passes to remove the length of the step, whilst cylindrical turning would require a lesser number of longer passes and therefore be more efficient. The cylindrical grinding, illustrates the concept of roughing, since it requires three stages of processing: rough and semi-finish turning and cylindrical grinding as the finishing step. Although grinding will result in a higher quality feature, the design has not specified that this is necessary. Therefore the process selection algorithm is able to select the faster alternative of cylindrical turning.

Table 8.4: Detailed process options for external step (IT 15)

feature	op	elem.	Tsetup	Tmachining	machine type	process
esp872162220	0	0	0.51	21.41	grinding_mc	rough_cyl_grinding
esp872162220	0	1	0.55	0.22	lathes	semi_cyl_turning
esp872162220	0	2	0.55	0.11	lathes	rough_cyl_turning
esp872162220	1	0	0.55	0.25	lathes	rough_profile_turning
esp872162220	2	0	0.55	0.21	lathes	rough_cyl_turning
esp872162220	3	0	0.55	0.41	lathes	rough_facing

For each operation element a machine tool type has been identified which is used for estimating times. If there are no machines of the required type available in the factory, then the process will be rejected before this stage. Note that the grinding option, operation 0, requires two machine tool types, the grinding machine and the lathe. This will result in the introduction of changeover costs, such as transportation and piece set-up costs, as well as batch set-up costs for both machines. The results further show that machining times for grinding are higher than for turning. The selected process of cylindrical turning requires 13 seconds of machining to produce the feature.

Table 8.5: Aggregate process plan for external step (IT 15)

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
A	Material						2.70	2.70
0	esp872162220	rough_cyl_turning	Mazak0	0.17	0.55	0.00	1.00	3.70
B	Material						2.70	2.70
0	esp872162220	rough_cyl_turning	Traub0	0.17	0.55	0.00	1.00	3.70
C	Material						2.70	2.70
0	esp872162220	rough_cyl_turning	Traub1	0.17	0.55	0.00	1.00	3.70

The selected operations, in this case cylindrical turning only, are passed on to the machine selection algorithm, which produces a number of alternative machine options, each of which is evaluated for cost and quality. The machine tool options are shown in Table 8.5. Clearly, since there is only one operation required, there is no need to determine a sequence. The estimated actual machining time for each feature has been calculated to be 10 seconds. This is less than the previous estimate of 13 seconds when the specific machine tool was not known. This is because the machine selected is better than the "average" machine tool data used for the initial estimate. In this example factory, however, the slower machines are not CNC lathes and were rejected for further consideration, thus the alternative machine tools which are shown each have the same cost. This is because the small geometry of the component prevents the full power of the machine tool from being used: the rate of cutting is constrained by the depth of cut instead of by the power. Also, all these machine tools have the same hourly cost rate. The cost per component is £3.71, which is divided between £2.70 for the raw material and £1 for the processing. Note that the set-up time for a component is three times as large as the processing time. Therefore, a large part of the processing cost is spent on non-productive time.

(b) The results of aggregate process planning for IT 13

To demonstrate the effects of quality specification, the tolerances for the feature were reduced. This should require higher quality processes. The tolerance interval was set to 0.2 mm, equivalent to a grade of IT=13. The results of the first stage, shown in Table 8.6, clearly show the effects of this tighter tolerance. It is no longer possible to use just rough turning, therefore two processing stages are needed. The system has selected semi-finish turning processes as the required level to finish the higher quality feature. However, it has determined that the depth of cut is too large for semi-finishing alone and therefore a roughing stage is required. This is to be performed using rough turning. The grinding alternative has not been affected by the increased quality, however, since it was already capable of achieving the new tolerance levels.

Table 8.6: Detailed process options for external step (IT 13)

feature	op	elem.	Tsetup	Tmachining	machine type	process
esp872162220	0	0	0.51	21.41	grinding_mc	rough_cyl_grinding
esp872162220	0	1	0.55	0.22	lathes	semi_cyl_turning
esp872162220	0	2	0.55	0.11	lathes	rough_cyl_turning
esp872162220	1	0	0.55	0.25	lathes	semi_profile_turning
esp872162220	1	1	0.55	0.12	lathes	rough_profile_turning
esp872162220	2	0	0.55	0.22	lathes	semi_cyl_turning
esp872162220	2	1	0.55	0.11	lathes	rough_cyl_turning
esp872162220	3	0	0.55	0.11	lathes	semi_facing
esp872162220	3	1	0.55	0.35	lathes	rough_facing

The requirement for two stages of processing has not affected the optimum process selection, however, since this is still cylindrical turning. The times shown indicate that the roughing process, requires 7 seconds to make a cut, whilst the semi-finishing cut will require 13.5 seconds. This reflects the less aggressive cutting conditions which are used in semi-finish turning in order to achieve the higher quality.

Table 8.7 shows the finished aggregate process plan generated from this intermediate data. The best route shows an overall cost of per unit of £3.85. Thus the increased quality requirement has resulted in an increased cost of £0.14 per unit. Whilst this is only an increase of 3.7% in total cost, the processing cost has increased by 14%. Routes A and B shown on the table use only a single machine tool. Therefore, the cost of setting up the machine is spread across all operation elements. In route C, however, two machines are specified. This is clearly a sub-optimal route, since it involves an unnecessary workpiece and machine set-up. This fact is reflected in an overall cost of £4.57, an increase of 19% in the cost. Since the processing times are the same, all this cost is due to the set-up and transfer requirements. The great majority of this cost will be from the set-ups, since the transfer times per unit can be seen to be up to two orders of magnitude lower, as in this example. Transfer costs are shared across the batch, so this time has been adjusted for the batch size (50 units), whilst piece set-up time is incurred for each part.

Table 8.7: Aggregate process plan for external step (IT 13)

index	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
A	Material						2.70	2.70
0	esp872162220	rough_cyl_turning	Mazak0	0.08	0.55	0.00	0.86	3.70
1	esp872162220	semi_cyl_turning	Mazak0	0.17	0.00	0.00	0.28	3.84
B	Material						2.70	2.70
0	esp872162220	rough_cyl_turning	Traub1	0.08	0.55	0.00	0.86	3.70
1	esp872162220	semi_cyl_turning	Traub1	0.17	0.00	0.00	0.28	3.84
C	Material						2.70	2.70
0	esp872162220	rough_cyl_turning	Traub0	0.08	0.55	0.00	0.86	3.70
1	esp872162220	semi_cyl_turning	Traub1	0.17	0.55	0.00	1.00	4.57

(c) The results of aggregate process planning for IT 10

A third planning run was performed for the same external step feature with even tighter tolerances, the diameter being set to an interval of 1mm (equivalent to IT=10). From Table 8.8, showing the process alternatives which were identified, it can be seen that each of the four process alternatives requires three stages of processing: a roughing stage, then a semi-finishing stage and then a finishing stage. For all processes except grinding, the same machine type is identified for each step. Grinding requires the use of two types of machine: a grinding machine for the finishing process (grinding) and a lathe for the roughing and semi-finishing steps (turning). This fact, plus the longer processing times for grinding, mean that the selected process is again cylindrical turning.

Table 8.8: Detailed process options for external step (IT 10)

feature	op	elem.	Tsetup	Tmachining	machine type	process
esp872162220	0	0	0.51	21.41	grinding_mc	rough_cyl_grinding
esp872162220	0	1	0.55	0.22	lathes	semi_cyl_turning
esp872162220	0	2	0.55	0.11	lathes	rough_cyl_turning
esp872162220	1	0	0.55	0.12	lathes	finish_profile_turning
esp872162220	1	1	0.55	0.25	lathes	semi_profile_turning
esp872162220	1	2	0.55	0.12	lathes	rough_profile_turning
esp872162220	2	0	0.55	0.12	lathes	finish_cyl_turning
esp872162220	2	1	0.55	0.22	lathes	semi_cyl_turning
esp872162220	2	2	0.55	0.11	lathes	rough_cyl_turning
esp872162220	3	0	0.55	0.05	lathes	finish_facing
esp872162220	3	1	0.55	0.11	lathes	semi_facing
esp872162220	3	2	0.55	0.35	lathes	rough_facing

In Table 8.9 three alternative process route are shown using the same cylindrical turning process. The first route is the one selected as the best option by the system, since it uses the same machine tool for each case and therefore set-up time is only added once. In the two alternative routes, a different machine tool is suggested for one of the operations. This unnecessary change causes an increase in set-up and transport cost, and makes these routes more expensive. The best route results in a total unit cost of £3.99, an increase of £0.14 on the lower quality example. The two alternative routes have costs of £4.71 for route B and £5.43 for route C. Although only two machine tools are required in each case, route C is more expensive, since it requires two machine changes, instead of just one.

Table 8.9: Alternative aggregate process plans for external step (IT 10)

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
A	Material						2.70	2.70
0	esp872162220	rough_cyl_turning	Mazak2	0.08	0.55	0.00	0.86	3.56
1	esp872162220	semi_cyl_turning	Mazak2	0.17	0.00	0.00	0.28	3.84
2	esp872162220	finish_cyl_turning	Mazak2	0.08	0.00	0.00	0.14	3.99
B	Material						2.70	2.70
0	esp872162220	rough_cyl_turning	Mazak0	0.08	0.55	0.00	0.86	3.56
1	esp872162220	semi_cyl_turning	Mazak2	0.17	0.55	0.01	0.28	3.84
2	esp872162220	finish_cyl_turning	Mazak2	0.08	0.00	0.00	0.86	4.71
C	Material						2.70	2.70
0	esp872162220	rough_cyl_turning	Mazak2	0.08	0.55	0.00	0.86	3.56
1	esp872162220	semi_cyl_turning	Mazak0	0.17	0.55	0.01	1.00	4.57
2	esp872162220	finish_cyl_turning	Mazak2	0.08	0.55	0.01	0.86	5.43

(d) Alternative process option

Whilst the previous examples have accepted the process selected in the first stage of the planning system, it is possible for the user to override the use of the best option and to pick a process with a higher cost. This will be an important requirement when two processes are close in costs, or when the best process is ruled out for other reasons. In Table 8.10, the consequences of selecting the grinding process instead of the cylindrical turning process at the process selection stage are shown. In this case, the requirement for two different classes of machine tool forces an extra machine tool change, and therefore the best result includes one extra set-up cost. This causes an increase in the overall unit cost, to £6.11.

Table 8.10: Aggregate process plan for external step (IT 10)

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						2.70	2.70
0	esp872162220	rough_cyl_turning	Mazak2	0.08	0.55	0.00	0.86	3.56
1	esp872162220	semi_cyl_turning	Mazak2	0.17	0.00	0.00	0.28	3.84
2	esp872162220	rough_cyl_grinding	Blanchard	21.41	0.37	0.00	35.39	39.24

8.4.2 Example 2: An internal cylindrical feature

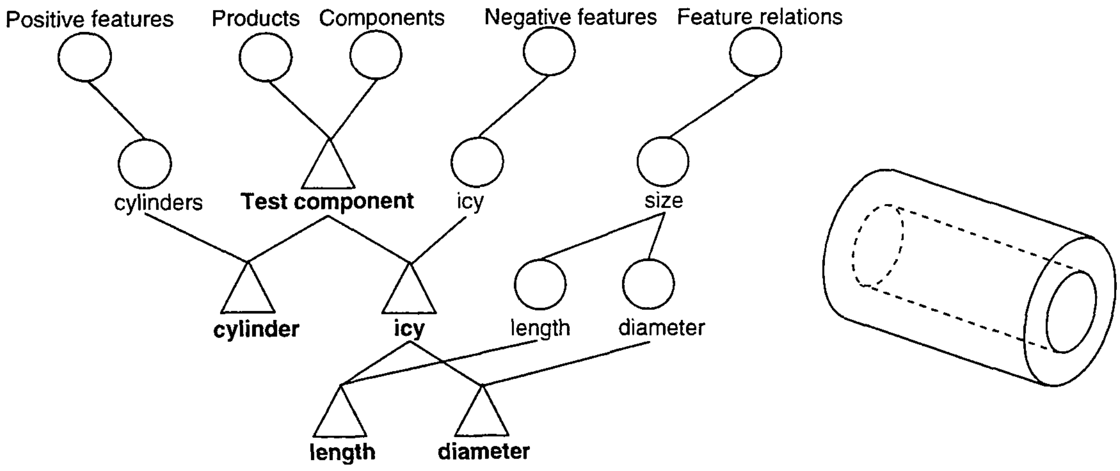


Figure 8.5: Internal cylinder feature example: product model

The second example presented is an internal cylindrical feature. In this instance, the aggregate process planning system must recognise the requirements for access for certain processes: for example, boring cannot be used to generate an internal diameter unless there is already an existing hole present. For this example, a component based upon a cylinder was again selected. Figure 8.5 shows the product model structure and a sketch of this component when completed. It can be seen that the structure is similar to the previous example of an external cylinder, with only the class of the negative feature being different. The properties of the feature and feature relation objects are detailed in Table 8.11. The feature selected is an internal cylinder, with a diameter of 40mm and a length of 50mm.

Table 8.11: Component with an internal cylindrical surface

Name	PARENTS	PROPERTIES	VALUES
Test_Component	components products	amount selfname typeclass material	1 Test_Component components mild steel
cylinder827426233	cylinder Test_Component	diameter length selfname typeclass	80.0 100.0 cylinder827426233 features

Name	PARENTS	PROPERTIES	VALUES
icy872162220	icy Moving_Armature	component number rank selfname set-up typeclass prefix it roughness	Test_Component 1 Unknown icy872162220 1 features icy Unknown Unknown
length_icy872162220	length icy872162220	upper lower nominal typeclass	0.5 0.5 50.0 geometry
diameter_icy872162220	diameter icy872162220	upper lower nominal typeclass	0.05 0.05 40.0 geometry

As for the previous example, versions of this feature of different quality specifications are described. Runs have been carried out for tolerance grades of IT 15 and IT 10. For an axi-symmetric component with an internal cylinder, the blank type selected could be any of solid billet, bar, tube billet and tube. The first two examples here consider the use of a solid billet whilst the third examines the tube billet alternative.

(a) Internal cylinder example (IT 15)

In this example, the blank type selected was the solid billet. The results in Table 8.12 show eight alternative processing methods. More options are identified for an internal cylinder than the external cylinder for two reasons: Firstly, there are simply more hole making processes available, such as drilling and reaming, that don't have equivalent external processes. Secondly, however, the hole making processes are in many cases specified with more than one machine type. Where this is the case, two operations are created, one for each machine type. Thus, in this example, where drilling is considered for an internal cylinder, it may be done on either a milling machine or a lathe. Drills are not considered since the hole is to be axi-symmetric, although milling machines are considered as accurate positioning is usually possible for these machine tools. The use of two alternative classes allows the system to consider at the process selection stage the best machine type to perform the operation on.

Table 8.12: Internal cylinder example. (IT 15, solid billet)

feature	op	elem.	Tsetup	Tmachining	machine type	process
icy872162220	0	0	0.73	0.60	lathes	drilling
icy872162220	0	1	0.68	42.82	grinding_mc	rough_cyl_grinding
icy872162220	0	2	0.73	0.24	lathes	semi_cyl_turning
icy872162220	1	0	0.70	1.10	milling_mc	drilling
icy872162220	2	0	0.70	0.98	milling_mc	drilling
icy872162220	2	1	0.70	0.57	milling_mc	reaming
icy872162220	3	0	0.70	0.98	milling_mc	drilling
icy872162220	3	1	0.70	0.22	milling_mc	rough_bore_milling
icy872162220	4	0	0.73	0.60	lathes	drilling
icy872162220	4	1	0.73	0.57	lathes	reaming
icy872162220	5	0	0.73	0.67	lathes	drilling
icy872162220	6	0	0.73	0.60	lathes	drilling
icy872162220	6	1	0.73	0.25	lathes	rough_cyl_boring
icy872162220	7	0	0.73	0.60	lathes	drilling
icy872162220	7	1	0.73	0.25	lathes	rough_profile_boring

In this case, the lathe appears to be the best option since the processing time is least. Using a lathe, Table 8.13 shows that a cost of £8.43 is estimated for the component. In contrast to the previous example of external turning, the processing time is similar in magnitude to the set-up time. This is because the amount of material to be removed is much greater in this case, since the hole is the full length of the workpiece (100mm) and cross-sectional area is greater.

Table 8.13: Drilling with a lathe (IT 15, solid billet)

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						6.54	6.54
0	icy872162220	drilling	Mazak2	0.60	0.73	0.00	1.91	8.43

If the milling machine alternative is selected from Table 8.12, then the machine tool selected will be a milling machine. This uses operation number 1 from Table 8.12 instead of operation number 5. The calculation of time uses the same equation, but the machine tool properties (i.e. maximum power, rpm and feed) may be different. Table 8.14 shows that, if a milling machine is used, the costs are shown to be £8.40. This is almost exactly the same as the cost calculated for the lathe. An examination of the time shows that the machining time is the same for both options, indicating that similar cutting parameters were assumed. Since the hole is of a relatively large diameter it is likely that the cutting velocity limited cutting speed in both cases. The difference in costs therefore arises from the difference in set-up times. The milling machine has a twin pallet set-up and therefore has a reduced set-up time compared to the lathe.

Table 8.14: Drilling with a milling machine (IT 15, solid billet)

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						6.54	6.54
0	icy872162220	drilling	AGV18a	0.60	0.70	0.00	1.88	8.40

(b) Example with high quality requirement (IT 10)

This example shows the consequences of increasing the quality specification. In this case, the tolerances on the depth and width of the feature have been tightened to IT 10. The tolerance on the length was not altered, since it will always be the full length of the component. A solid billet was again used as the blank workpiece shape.

Table 8.15: Internal cylinder example (IT 10, solid billet)

feature	op	elem.	Tsetup	Tmachining	machine type	process
icy872162220	0	0	0.73	0.60	lathes	drilling
icy872162220	0	1	0.68	42.82	grinding_mc	rough_cyl_grinding
icy872162220	0	2	0.73	0.24	lathes	semi_cyl_turning
icy872162220	1	0	0.70	0.98	milling_mc	drilling
icy872162220	1	1	0.70	0.57	milling_mc	reaming
icy872162220	2	0	0.70	0.98	milling_mc	drilling
icy872162220	2	1	0.70	0.22	milling_mc	semi_bore_milling
icy872162220	3	0	0.73	0.60	lathes	drilling
icy872162220	3	1	0.73	0.57	lathes	reaming
icy872162220	4	0	0.73	0.60	lathes	drilling
icy872162220	4	1	0.73	0.25	lathes	finish_cyl_boring
icy872162220	4	2	0.73	0.25	lathes	semi_cyl_boring

The results in Table 8.15 show that for the higher quality version of the feature, the alternative processes are different from the low quality version. It is no longer possible to use just one process step (i.e. drilling) to make the feature, since this would not result in sufficiently high tolerances. Therefore, a finishing process is required. The effect of this is to remove the drilling only options and reduce the number of options to five. The higher quality options which were available at the previous design quality are still available, so the system must select one of these. Table 8.15 shows that the selected process was bore milling. The alternative processes of cylindrical boring and reaming were also investigated.

Table 8.16 shows the results of the machine selection for bore milling. In this process, an edge milling cutter is moved around the cylinder to widen an initial hole. For deep holes, helical interpolation can be used to overcome limits in axial depth of cut. This process requires an initial drilling operation, as shown. This can be carried out on the milling machine, so no machine change is required. The time for bore milling is lower than that for turning, as more than one cutting edge may be removing material at any one time. The total cost calculated for this example is £8.51, an increase of £0.11 on the lower quality example. The additional milling process has added an extra cost, whilst the drilling cost is slightly reduced as the material removed is less.

Table 8.16: Finishing the internal cylinder using bore milling (solid billet)

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						6.54	6.54
0	icy872162220	drilling	AGV18a	0.53	0.70	0.00	1.77	8.31
1	icy872162220	semi_bore_milling	AGV18a	0.12	0.00	0.00	0.19	8.51

Table 8.17: Finishing the internal cylinder using boring (solid billet)

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
A	Material						6.54	6.54
0	icy872162220	drilling	Mazak2	0.17	0.73	0.00	1.22	7.76
1	icy872162220	semi_cyl_boring	Mazak2	0.53	0.00	0.00	0.85	8.62
2	icy872162220	finish_cyl_boring	Mazak2	0.17	0.00	0.00	0.27	8.90
B	Material						6.54	6.54
0	icy872162220	drilling	Multidrill	0.17	0.73	0.00	1.22	7.76
1	icy872162220	finish_cyl_boring	Mazak2	0.53	0.73	0.01	1.22	8.99
2	icy872162220	semi_cyl_boring	Mazak2	0.17	0.00	0.00	0.85	9.85

Boring, shown in Table 8.17, requires three stages to the process, since finish and semi-finish boring is required after turning. Finish cylindrical boring is not suitable for cutting directly after drilling since the surface conditions are usually not suitable. This may be for reasons of roughness or because the outer layer of material has been hardened by the primary production process. Thus, it is necessary to schedule an extra stage of semi-finish turning to prepare the surface. The best cost calculated for boring is £8.90, for route A which uses a single machine tool, which represents an increase of £0.50 on drilling alone. This is a 27% increase in processing costs which clearly indicates the cost penalty of requiring multiple processes. Route B shows that the costs are even higher if a machine tool change is required, with a cost of £9.85. This example specifies a specialist drilling machine for the initial hole.

The reaming process is planned with just two stages: since reaming tools are available in only standard sizes, if a reaming tool can be used, there will be an equivalent sized drill to create the initial hole. CESS does not currently include a mechanism to check the diameter of a hole to determine if it is a standard size for reaming and drilling. This would be a simple additional check that could be added if a database of standard tools was available. The results of machine selection for reaming, shown in Table 8.18, identify a lathe to perform both operations. The total cost in this instance is £9.26, which is higher than both boring and mill boring options.

Table 8.18: Finishing the internal cylinder using reaming (solid billet)

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						6.54	6.54
0	icy872162220	drilling	Mazak0	0.53	0.73	0.00	1.80	8.34
1	icy872162220	reaming	Mazak0	0.57	0.00	0.00	0.91	9.26

(c) Example with hollow blank

The other option for a hollow part is to use a tube billet, so that the blank workpiece starts with an internal diameter. In this case, less material must be removed since the internal cylinder merely requires widening to the design diameter. Drilling will no longer be required as a roughing step for many of the processes. The example tested here used the same product model as for the previous examples, i.e. a cylinder with an internal bore of diameter 40mm and a tolerance grade of IT 10.

Table 8.19: Internal cylinder (IT 10, tube billet)

feature	op	elem.	Tsetup	Tmachining	machine type	process
icy872162220	0	0	0.51	42.82	grinding_mc	rough_cyl_grinding
icy872162220	0	1	0.55	0.24	lathes	semi_cyl_turning
icy872162220	1	0	0.54	0.22	milling_mc	semi_bore_milling
icy872162220	2	0	0.55	0.25	lathes	finish_cyl_boring
icy872162220	2	1	0.55	0.25	lathes	semi_cyl_boring

With a tube billet blank selected, the process options can be seen to have changed from the same feature using a solid billet. In this case, there are only three options. However, each of these options requires fewer steps than the equivalent previous case. The selection is between grinding, boring and bore milling processes. Of these, bore milling can create the feature in a single stage, whilst the other processes require preliminary turning. The reaming options have been removed from the plan because the depth of cut from the raw material to the final diameter is too large. Reaming would require an additional drilling or boring process before it could be used. The process selection option again suggests bore milling because of the reduced cost compared to the alternatives. Results of the machine selection, shown in Table 8.20, show a cost of £6.24. It is important to not that the bore milling process is operating at close to the maximum depth for the feature diameter: tools which are sufficiently slender may not be available in all factories. The tool limits which are set by a particular user of the system might preclude the use of this process in this instance.

Table 8.20: Finishing external cylinder with bore milling (tube billet)

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						5.33	5.33
0	icy872162220	semi_bore_milling	AGV18a	0.12	0.54	0.00	0.90	6.24

The alternative to bore milling is to use a boring process. This is shown as operation number 2 on Table 8.20. Two processing stages are required to achieve the required surface finish. Just as finish boring cannot be used directly after drilling, it is unsuitable for use on raw material, which may have been hardened by the primary forming process. Thus, a semi-finishing pass must be scheduled. Each process requires 10 seconds for a cutting pass, resulting in a total cost of £6.61, which is sufficiently close to the value predicted for bore milling that this could be used as an alternative. This would be particularly likely if further axi-symmetric features were present, or if a coaxiality tolerance was added to the feature.

Table 8.21: Finishing external cylinder with boring (tube billet)

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						5.33	5.33
0	icy872162220	semi_cyl_boring	Mazak0	0.17	0.55	0.00	0.99	6.33
1	icy872162220	finish_cyl_boring	Mazak0	0.17	0.00	0.00	0.27	6.61

8.4.3 Example 3: A prismatic feature

The previous examples have been axi-symmetric components, a third example is therefore shown to illustrate the functioning of the planning system for prismatic processes. As in the previous examples, a simple product consisting of a positive feature (in this case a prism) and a single negative feature was defined. The product model structure for this component is shown in Figure 8.6. Once again, the structure of the model is similar to the previous examples. The main difference in this case is that an extra feature relation is required since the feature is defined by three dimensions instead of two. Each of these dimensions belongs to sub-classes of the size class of feature relations. The class to which the positive feature is attached has also changed, in this case to the prism class.

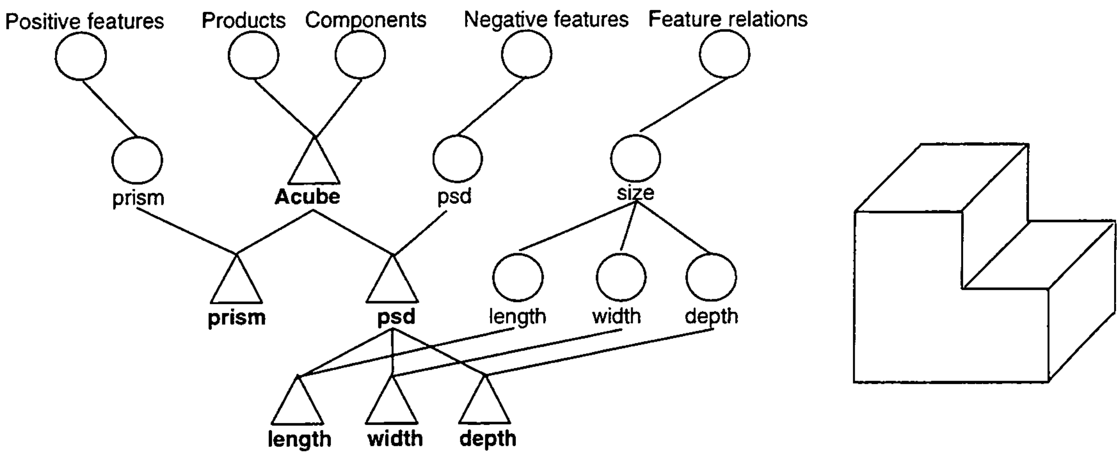


Figure 8.6: Prismatic feature component: product model

The negative feature selected for this example is a square shoulder (*psd*). This feature, illustrated in Figure 8.6, is one of the more simple prismatic features. The details of the attributes of the product model objects is set out in Table 8.22. The properties of the feature object (*psd*) show that quality specifications have been set at the feature level instead of the feature relation level. This indicates that the approximate quality level of the feature is known, but detailed planning of the dimensions of the component has yet to be done.

Table 8.22: A component with a prismatic feature

Name	PARENTS	PROPERTIES	VALUES
acube	components products	amount selfname typeclass material	1 acube components mild steel
psd875102798	psd acube	component number selfname set-up typeclass prefix it roughness	acube 1 psd875102798 1 features psd 14 20.0
depth_psd875102798	depth psd875102798	upper lower nominal typeclass	Unknown Unknown 30.0 geometry
width_psd875102798	width psd875102798	upper lower nominal typeclass	Unknown Unknown 30.0 geometry
length_psd875102798	length psd875102798	upper lower nominal typeclass	0.5 0.5 200.0 geometry
prism875102774	prism acube	length width breadth selfname typeclass	200.0 200.0 200.0 prism875102774 features

(a) Results of planning for square shoulder (IT 14)

As shown in Table 8.23, the system identifies three alternative processes for the creation of the square shoulder feature, flat grinding, shoulder milling and contouring. In this instance, the two milling processes are effectively the same, since shoulder milling can be considered a special case of contouring. Grinding requires initial milling operations as roughing stages, whilst the milling processes can produce the required depth of cut in one operation element. The shouldering process is selected by the system. Table 8.24 shows the results of machine selection for the component. There is only one milling machine available in the factory model, so alternative options are not available. Note that the overall cost for this component of £102.63 includes a substantial material cost. The actual cost of processing, exclude set-up costs is calculated at £6.24.

Table 8.23: Process alternatives for square shoulder

feature	op	elem.	Tsetup	Tmachining	machine type	process
psd875102798	0	0	2.67	0.91	grinding_mc	rough_flat_grinding
psd875102798	0	1	3.65	0.21	milling_mc	semi_face_milling
psd875102798	0	2	3.65	1.44	milling_mc	rough_face_milling
psd875102798	1	0	3.65	1.70	milling_mc	rough_contouring
psd875102798	2	0	3.65	1.70	milling_mc	rough_shoulder_milling

Table 8.24: Aggregate process plan for square shoulder

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						96.39	96.39
0	psd875102798	rough_shoulder_milling	AGV18a	0.93	3.66	0.00	6.24	102.63

(b) Results of planning for square shoulder (IT 10)

A second run of the aggregate process planning system was undertaken for this feature using a higher quality specification. To do this, tolerances were defined for individual dimensions of the feature. In this case a tolerance interval of 0.06 mm for both the depth and the width of the feature was set by altering the feature relation properties. The results of process selection, shown in Table 8.25, show that in this case the feature quality specified is high enough to require a two stage process, with semi-finishing and roughing being applied.

Table 8.25: Process operations of square shoulder feature (IT 10)

feature	op	elem.	Tsetup	Tmachining	machine type	process
psd875102798	0	0	2.67	0.91	grinding_mc	rough_flat_grinding
psd875102798	0	1	3.65	0.21	milling_mc	semi_face_milling
psd875102798	0	2	3.65	1.44	milling_mc	rough_face_milling
psd875102798	1	0	3.65	0.32	milling_mc	semi_contouring
psd875102798	1	1	3.65	1.38	milling_mc	rough_contouring
psd875102798	2	0	3.65	0.21	milling_mc	semi_shoulder_milling
psd875102798	2	1	3.65	1.49	milling_mc	rough_shoulder_milling

The results of machine selection for the higher quality specification feature, shown in Table 8.26, indicate that the system predicts a the same cost as for the previous example. This is because the number of machining passes does not change. The only difference is the tools which are to be used. and that the final pass will have a different (lesser) depth of cut to the previous passes.

Table 8.26: Machine selection result for milling square shoulder

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						96.39	96.39
0	psd875102798	rough_shoulder_milling	AGV18a	0.81	3.66	0.00	6.06	102.44
1	psd875102798	semi_shoulder_milling	AGV18a	0.12	0.00	0.00	0.19	102.63

Another run was performed to investigate the alternative process options further. In this case, the grinding process was selected, which is operation number 0 in Table 8.25. The results of the machine selection algorithm for this operation are shown in Table 8.27. The cost using grinding is £107, an increase of £5 on milling alone.

Table 8.27: Machine selection result for grinding square shoulder

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						96.39	96.39
0	psd875102798	rough_face_milling	AGV18a	0.79	3.66	0.00	6.01	102.40
1	psd875102798	semi_face_milling	AGV18a	0.12	0.00	0.00	0.19	102.58
2	psd875102798	rough_flat_grinding	Keithley	0.91	2.68	0.02	5.20	107.78

8.4.4 Example 4: A component with two features

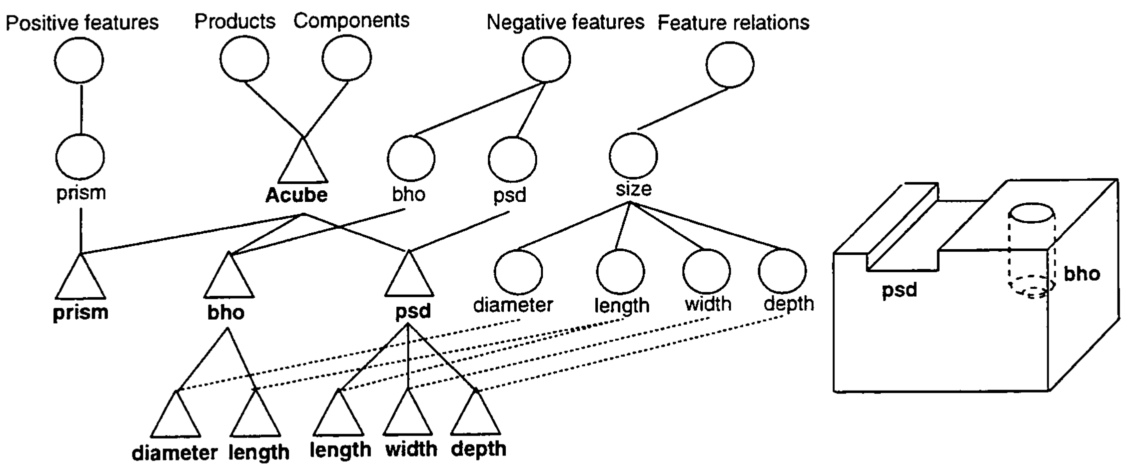


Figure 8.7: Component with two features

The examples in this section have demonstrated the successful functioning of the aggregate process planning system on different cases of single features. To be useful as a concurrent engineering tool, however, the system must be able to develop process plans for many features at the same time. Further examples are shown using components with two and three features. Tests to investigate the performance of the system with components having a larger number of features are described in the next section.

This example uses a simple prismatic component with two features. The same overall dimensions are used as in the previous example, i.e. 200mm length, depth and width. The features which are used are a prismatic slot (*pst*) and a prismatic blind hole (bho). The blind hole has no requirement to be square at the bottom, so conventional drilling processes can be considered. The slot has no additional geometry specified over the minimum, so it is considered a square slot. The component is shown in Figure 8.7, together with the product model, the full details of which are shown in Appendix E, Table 3.

feature	op	elem.	Tsetup	Tmachining	machine type	process
bho875574912	0	0	3.65	0.09	milling_mc	drilling
bho875574912	0	1	3.65	0.09	milling_mc	reaming
bho875574912	1	0	3.65	0.09	milling_mc	drilling
bho875574912	2	0	3.65	0.09	milling_mc	drilling
bho875574912	2	1	3.65	0.01	milling_mc	rough_bore_milling
bho875574912	3	0	3.65	0.09	drills	drilling
bho875574912	4	0	3.65	0.09	drills	drilling
bho875574912	4	1	3.65	0.09	drills	reaming
pst875574810	0	0	3.65	0.35	milling_mc	semi_cavity_milling
pst875574810	0	1	3.65	0.71	milling_mc	rough_cavity_milling
pst875574810	1	0	3.65	0.35	milling_mc	semi_slot_milling
pst875574810	1	1	3.65	0.71	milling_mc	rough_slot_milling

The first stage of the plan now shows operations for both features. The first five of the operations are for the hole, the remaining two for the slot. The process options for the hole are: reaming, drilling and bore milling. The first two are considered in two cases- those on simple drilling machines, and those on milling machines. The bore milling must be done on a milling machine. The reaming and bore milling operations require preliminary drilling, whilst using drilling alone is a single step. The slotting options are cavity milling and slotting. Both require a roughing and finishing step. Cavity milling is suggested, even though access is available, since it can clearly make the feature. Slotting will be expected to be favoured, as the specialist process. The blind hole is two small to be made with cavity milling, otherwise the two might have been made using a single process type. The process selection chooses to use the milling version of drilling, so that only one machine type is required. Slotting is preferred over cavity milling as expected.

Table 8.28: Machine selection results for prismatic component

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						96.38	96.38
0	bho875574912	drilling	AGV18a	0.04	3.65	0.00	4.82	101.21
1	pst875574810	rough_slot_milling	AGV18a	0.38	0.00	0.00	0.62	101.83
2	pst875574810	semi_slot_milling	AGV18a	0.19	0.00	0.00	0.31	102.14

The machine selection output, shown in Table 8.28, consists of one route, since there is only one milling machine active in the factory model. The times show that the slot feature requires the majority of the processing time. The major costs are shown to be the material cost and the set-up costs, since this is a large component. The difference in costs between features can used by the designer to search for cost savings.

8.4.5 Example 5: A component with three features

The previous example used a component design where access to both features was possible from the same set-up. This example describes the planning for a component where two set-ups will be required. The component, shown in Figure 8.8, has three negative features: an internal diameter, an internal thread and an external step. This component has been deliberately designed to test the set-up functionality. Since the thread on the internal bore and the external step are at different ends of the component, these cannot be machined together in the same set-up. To indicate this in the product model, the value of the set-up property attached to these features must be different. The internal bore and the thread are grouped together in one set-up by giving them the same value (set-up = 1), whilst the external step is assigned to a different set-up (set-up = 2). The full details of this component are shown in Appendix E, Table 4.

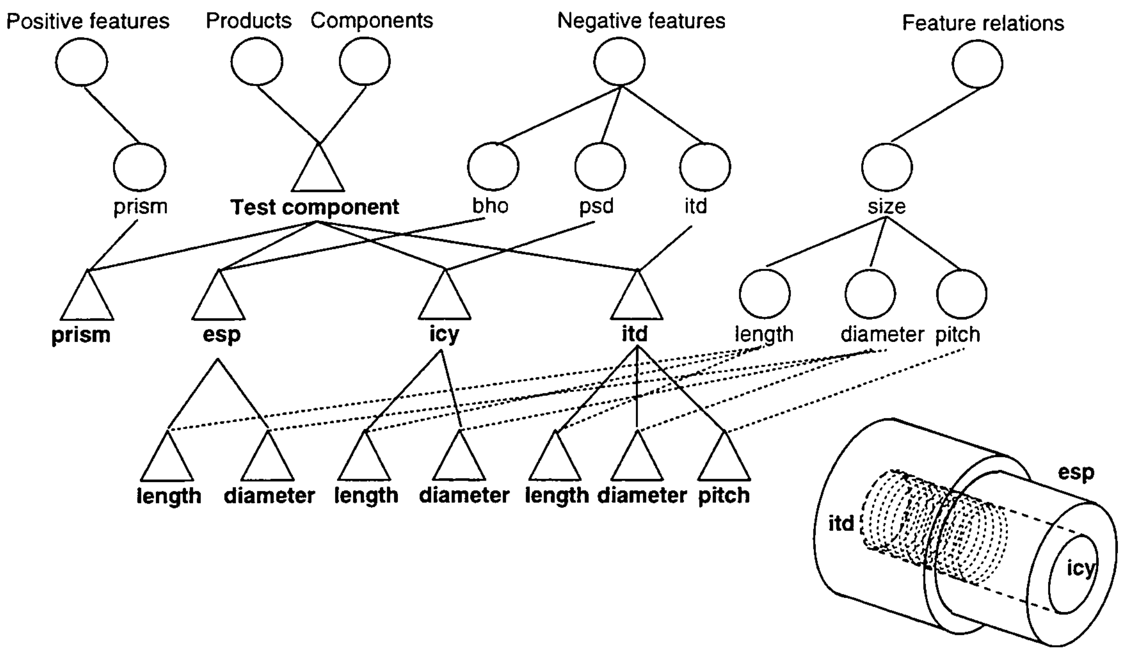


Figure 8.8: Cylindrical component with three features

Table 8.29: Process options for three feature example

feature	op	elem.	Tsetup	Tmachining	machine type	process
itd875577220	0	0	0.54	9.99	milling_mc	thread_milling
itd875577220	1	0	0.55	0.07	lathes	thread_boring
icy875576021	0	0	0.51	31.41	grinding_mc	rough_cyl_grinding
icy875576021	0	1	0.55	0.23	lathes	semi_cyl_turning
icy875576021	1	0	0.54	0.11	milling_mc	semi_bore_milling
icy875576021	2	0	0.55	0.25	lathes	finish_cyl_boring
icy875576021	2	1	0.55	0.25	lathes	semi_cyl_boring
esp872162220	0	0	0.51	31.41	grinding_mc	rough_cyl_grinding
esp872162220	0	1	0.55	0.22	lathes	semi_cyl_turning
esp872162220	0	2	0.55	0.11	lathes	rough_cyl_turning
esp872162220	1	0	0.55	0.25	lathes	rough_profile_turning
esp872162220	2	0	0.55	0.21	lathes	rough_cyl_turning
esp872162220	3	0	0.55	0.41	lathes	rough_facing

The results of the process identification and evaluation stage show a number of alternatives for each feature. The processes available, by feature, are:

itd thread boring and thread milling (hole is too large for tapping);

icy bore milling, grinding, boring;

esp turning, profile turning and facing.

All the operation elements selected share the same machine type, the lathe, so the same machine tool can potentially be used for all the processing.

Table 8.30: Alternative routes for the three feature example

index	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
A	Material						2.46	2.46
0	icy875576021	semi_cyl_boring	Mazak0	0.08	0.55	0.00	0.85	3.32
1	icy875576021	finish_cyl_boring	Mazak0	0.08	0.00	0.00	0.13	3.45
2	itd875577220	thread_boring	Mazak0	0.07	0.00	0.00	0.11	3.57
3	esp872162220	rough_cyl_turning	Mazak0	0.17	0.55	0.00	1.00	4.57
B	Material						2.46	2.46
0	icy875576021	semi_cyl_boring	Mazak2	0.11	0.55	0.00	0.89	3.36
1	icy875576021	finish_cyl_boring	Mazak2	0.11	0.00	0.00	0.17	3.54
2	itd875577220	thread_boring	Mazak2	0.07	0.00	0.00	0.11	3.65
3	esp872162220	rough_cyl_turning	Mazak2	0.17	0.55	0.00	1.00	4.66

Table 8.30 shows the results of machine selection for this component. The successful functioning of the set-up algorithm is shown in the value of *tp* for the fourth operation. This is non-zero even though the machine tool has not changed, because the feature is forced to be in a different set-up from the previous features. Two alternative routes are shown, each using a single machine tool. Note that the boring operations have different speeds on these two machines. This indicates the difference between the maximum spindle speeds which is the limiting factor for the process in this instance.

8.5 Testing the system with real products

8.5.1 Case study: Moving armature component

This case study shows the aggregate process planning results of the moving armature component. The product model for this component is shown in Figure 8.1. The aggregate process planning system was run for this component product model using the Warner Electric factory model. The first stage of this process was to identify potential processes. Once these processes have been identified, the system performs an evaluation of production times for each step of the process. These times are then used to select the best process option for each feature. The combined results of these functions are shown in Table 8.31. For each feature, one or more alternative operations have been generated. It can be seen that a wide range of processes have been identified for the various features of this component. Processes belonging to the classes of turning, milling, grinding and drilling have been identified. The process alternatives identified for each feature will be discussed in turn.

The first feature is the prismatic surface (*sf2*), which represents two notches on the component. Contouring has been identified as the only process capable of machining this feature, so this operation must be in the route. This feature, which can be machined using a single pass of an edge milling tool, requires a milling machine. For both the external face (*efa*) features, two alternatives have been identified: flat grinding and facing, requiring a grinding machine and a lathe respectively. Each option requires only a single processing step, without previous roughing. The process selection algorithm must choose one of these operations for the route. Facing is seen to be a faster option than grinding. Five finishing operations have been identified for the internal bore (icy), all of which require both a roughing and finishing process. Since the bore is made from solid material, every process requires an initial drilling process. The alternatives are: cylindrical grinding, reaming, bore milling and cylindrical boring. The reaming option is divided into two operations, using either a milling machine or a lathe, to create the fifth option. Bore milling and reaming require only the drilling process as a roughing step, whilst grinding and boring both require an additional intermediary stage of semi-finish boring. Five processing options have been identified for the prismatic hole (pho) feature, representing the three holes in the component. These holes are to a lower tolerance than the bore, so the processes suggested include lower quality options. The options identified are drilling, reaming and bore milling. Of these, reaming and bore milling require roughing using a drilling operation. Thus, drilling alone is clearly the better option. The system has selected drilling on a milling machine instead of a drill to minimise the number of set-ups.

Whilst four machine types have been identified as possible options, the minimum number which can be used is two. Only milling machines and lathes are required for the selected combination of operations, since the drilling is to be carried out on the milling machine. The selection algorithm will tend to select the processes which result in the lowest number of machines being used, as in this case. The operation elements do not show a great variation in set-up times, which is expected as the same component is to be handled throughout. The importance of the set-up times is that it penalises the selection of multiple processes where a single process could be used. Finally, two alternatives have been identified to finish the external cylinder (ecy) feature, grinding and turning. Since grinding is slower and requires a roughing stage, the faster turning option has been chosen.

Table 8.31: Process options for moving armature

feature	op	elem.	Tsetup	Tmachining	machine type	process
sf2872166221	0	0	0.543	0.017	milling_mc	semi_contouring
efa872166099	0	0	0.515	0.609	grinding_mc	rough_face_grinding
efa872166099	1	0	0.555	0.128	lathes	finish_facing
icy827756050	0	0	0.555	0.023	lathes	drilling
icy827756050	0	1	0.515	2.921	grinding_mc	rough_cyl_grinding
icy827756050	0	2	0.555	0.014	lathes	semi_cyl_turning
icy827756050	1	0	0.543	0.034	milling_mc	drilling
icy827756050	1	1	0.543	0.027	milling_mc	reaming
icy827756050	2	0	0.543	0.034	milling_mc	drilling
icy827756050	2	1	0.543	0.026	milling_mc	semi_bore_milling
icy827756050	3	0	0.555	0.023	lathes	drilling
icy827756050	3	1	0.555	0.027	lathes	reaming
icy827756050	4	0	0.555	0.023	lathes	drilling
icy827756050	4	1	0.555	0.015	lathes	finish_cyl_boring
icy827756050	4	2	0.555	0.015	lathes	semi_cyl_boring
pho827756091	0	0	0.543	0.041	milling_mc	drilling
pho827756091	0	1	0.543	0.041	milling_mc	reaming
pho827756091	1	0	0.543	0.041	milling_mc	drilling
pho827756091	2	0	0.543	0.041	milling_mc	drilling
pho827756091	2	1	0.543	0.021	milling_mc	rough_bore_milling
pho827756091	3	0	0.543	0.041	drills	drilling
pho827756091	4	0	0.543	0.041	drills	drilling
pho827756091	4	1	0.543	0.041	drills	reaming
ecy827756181	0	0	0.515	9.517	grinding_mc	rough_cyl_grinding
ecy827756181	0	1	0.555	0.027	lathes	semi_cyl_turning
ecy827756181	1	0	0.555	0.027	lathes	rough_cyl_turning
efa872162220	0	0	0.515	0.609	grinding_mc	rough_face_grinding
efa872162220	1	0	0.555	0.128	lathes	semi_facing

Once the processes have been selected, the aggregate process planning system moves into its next phase. In this phase, the system generates the set of possible machine tools for each process and evaluates the cost for each machine tool. The cost and times for each of the operations is calculated for each machine option. Elements are then ordered into a feasible manufacturing sequence, before the second genetic algorithm selects best combination of machines for the component as a whole. The second genetic algorithm produces several alternative routings, which are ranked in order, that may be browsed through the use of the alternative route window, shown in Figure 8.9.

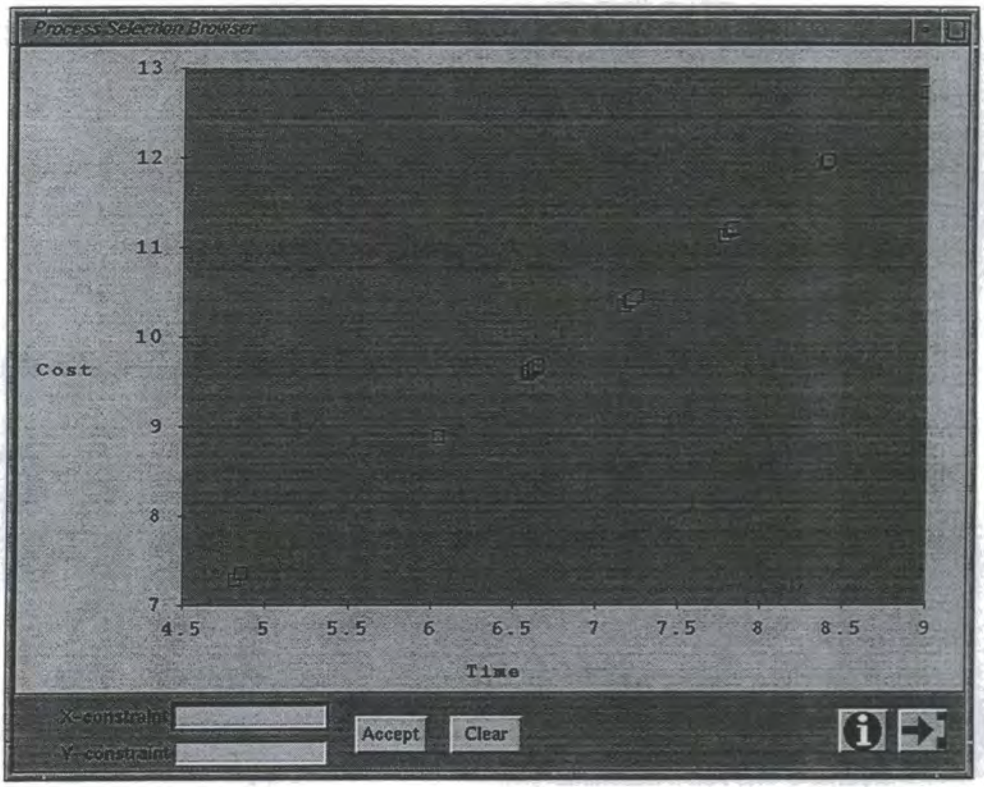


Figure 8.9: Alternative process plans shown graphically

This window displays the routes graphically as points on a time against cost graph. This graph is designed to allow the user to quickly browse through the alternative outputs. The system has the ability to store up to thirty of the best alternative routes found during the machine selection algorithm. The time graph concept allows the routes to be sorted in order of two manufacturability indicators at once. In this case, the criteria of processing time and component cost are shown. A further development of this method would include the lead time and the quality indicators as options. The points on the graph shown are approximately linear, as is expected since the cost is proportional to the time when the machine tool cost rate remains unchanged, as in these examples. Provision is made for limits such as maximum cost to be plotted on the chart as an aid to designers. A selection of these alternative routes which were produced for the moving armature is shown in Table 8.32.

Table 8.32: Alternative process routes for moving armature

index	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
A	Material						0.982	0.982
0	ecy827756181	rough_cyl_turning	Traub1	0.026	0.555	0.000	0.764	1.746
1	efa872162220	semi_facing	Traub1	0.042	0.000	0.000	0.068	1.815
2	efa872166099	finish_facing	Traub1	0.042	0.555	0.000	0.789	2.605
3	icy827756050	drilling	AGV18a	0.023	0.543	0.008	0.744	3.349
4	icy827756050	semi_bore_milling	AGV18a	0.012	0.000	0.000	0.019	3.368
5	pho827756091	drilling	AGV18a	0.021	0.000	0.000	0.034	3.403
6	sf2872166221	semi_contouring	AGV18a	0.009	0.000	0.000	0.014	3.418
B	Material						0.982	0.982
0	ecy827756181	rough_cyl_turning	Mazak2	0.026	0.555	0.000	0.764	1.746
1	efa872162220	semi_facing	Mazak2	0.056	0.000	0.000	0.091	1.838
2	efa872166099	finish_facing	Mazak2	0.056	0.555	0.000	0.812	2.650
3	icy827756050	drilling	AGV18a	0.023	0.543	0.003	0.743	3.394
4	icy827756050	semi_bore_milling	AGV18a	0.012	0.000	0.000	0.019	3.413
5	pho827756091	drilling	AGV18a	0.021	0.000	0.000	0.034	3.448
6	sf2872166221	semi_contouring	AGV18a	0.009	0.000	0.000	0.014	3.463
c	Material						0.982	0.982
0	ecy827756181	rough_cyl_turning	Mazak2	0.026	0.555	0.000	0.764	1.746
1	efa872162220	semi_facing	Mazak0	0.036	0.555	0.016	0.781	2.528
2	efa872166099	finish_facing	Mazak2	0.056	0.555	0.016	0.814	3.343
3	icy827756050	drilling	AGV18a	0.023	0.543	0.003	0.743	4.086
4	icy827756050	semi_bore_milling	AGV18a	0.012	0.000	0.000	0.019	4.106
5	pho827756091	drilling	AGV18a	0.021	0.000	0.000	0.034	4.140
6	sf2872166221	semi_contouring	AGV18a	0.009	0.000	0.000	0.014	4.155

Route A is the best route which was identified by the aggregate process planning systems. This table shows a number of the system outputs. The first piece of information which can be obtained in the production sequence, since this plan has been ordered into a fixed sequence. The system has determined that the turning processes should be carried out before the milling processes. In addition, the individual elements in the operations have been sorted according to the principle of roughing first, finishing last. In addition to the production sequence, the table shows a breakdown of processing cost, set-up time (*tp*) and processing time (*tm*) for each of the operations. The machine entry shows which machine tool has been selected to perform the operation and the *tt* column shows the transfer time to move between machines (in minutes per unit). The total cost of the component is shown as a running total (sum), which includes both machine and workpiece set-up times, material costs and quality costs as well as the sum of the processing costs. It can be seen that the best route has been identified as one which uses just two machine tools, one from each class, in order to minimise transfer and set-up costs. The total cost calculated is £3.42 for this route. Notice that the processing on the lathe is divided into two set-ups, as indicated by the addition of the set-up cost twice, one in element 0 and again in element 2. This is a consequence of the requirement to face both ends of the cylinder. The first alternative route, B, is very similar, only differing in the use of an alternative lathe, which has slightly higher costs.

Routes C is less suitable, with higher costs, caused by the use of more than two machine tools. This is an example of the routes which the selection algorithm can consider. This illustrates the ability to correctly assess the costs of alternative routing strategies. Depending on the machine tools available and the component design, it is possible that the best route will involve visits to multiple machine tools, instead of just a single machine. An example would be where a specialist machine such as a drill could outperform a less powerful general machine tool for a large feature. The aggregate process planning methodology will find these examples since it evaluates all options equally, instead of applying rules which may have exceptions.

Whilst the previous examples used a milling process to generate the internal diameter of the cylinder, an alternative process option would be to use cylindrical boring. This involves replacing operation number 2 for the *icy* feature with operation number 4. The results of the machine selection for this alternative are shown in Table 8.33. The total cost calculated for this route is £4.13, and increase of £0.60 on the best route.

Table 8.33: Process plan for moving armature using boring

index	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						0.982	0.982
0	icy827756050	drilling	Mazak0	0.023	0.555	0.000	0.758	1.740
1	icy827756050	semi_cyl_boring	Mazak0	0.007	0.000	0.000	0.012	1.753
2	icy827756050	finish_cyl_boring	Mazak0	0.007	0.000	0.000	0.012	1.765
3	ecy827756181	rough_cyl_turning	Mazak0	0.026	0.000	0.000	0.042	1.807
4	efa872166099	semi_facing	Mazak0	0.036	0.000	0.000	0.780	2.589
5	efa872162220	finish_facing	Mazak0	0.036	0.000	0.000	0.780	3.369
6	pho827756091	drilling	AGV18a	0.021	0.543	0.013	0.742	4.112
7	sf2872166221	semi_contouring	AGV18a	0.009	0.000	0.000	0.014	4.126

It is worth comparing the aggregate process routes generated with the actual detailed production plans that the company uses to create this component. In this instance, the actual process plan calls for the component to be made on three machine tools:

1. A lathe is used to turn the billet to the correct diameter,
2. The internal bore, the holes and the notch (*sf2*) are machined on a machining centre and,
3. A surface grinding machine is used to grind the faces of the component flat and to ensure the correct component thickness.

The available cost breakdown from the company indicates that total manufacturing cost is £1.41 per unit, of which material cost is £1.05. It is important to note, however, that this cost figure is highly dependent on the costing method applied by the company. This cost figure uses a different valuation of machining time, which is costed by the company only according to labour use. The CESS model uses a more accurate costing since it includes the cost of machine time according to depreciation. In this case, the cost value of machine time in CESS is approximately ten times that by the company, although it varies according to the machine. Using this costing method, a revised estimate of the component cost using the actual cutting times results in a cost of £4.65, which is comparable to the CESS result of £3.40.

8.5.2 Case study: Flange component

The flange example, shown previously in Figure 8.2, is a more complex product than the inductor, having more features. It is also a much larger component which does not fit into the machine tools in the Warner Electric factory. A second factory has therefore been modelled for testing larger components. This consists of two cells, containing lathes, machining centres and a vertical boring machine. A full breakdown of this factory model is shown in Appendix X. The blank type selected for this component during process planning was the tube billet, which is the only appropriate one since the internal diameter is so large. To use a solid billet would result in the removal of most of the initial volume of material. Table 8.34 shows the evaluated process alternatives together with the results of the selection from the first genetic algorithm (selected operations are highlighted). Multiple process options have been generated for each feature. Note, however, than in most cases only a single process step is required, since the quality specification of the component is generally low. The only exceptions to this rule are the reaming operations, which require a roughing process to create the initial hole in the material. Features of particular interest include the internal thread feature.

This is a thread on a large diameter bore (d=295mm), so that of the thread making processes in the CESS process model, only thread boring and thread milling can be used. Thread making using a solid tapping tool is not feasible since taps are not made of this size and since the thread is non-standard. The selection of drilling on a milling machine instead of on a drill for the blind holes is appropriate since there are twelve instances of this feature, each of which must be positioned accurately. To do this on a manually positioned drill would take much longer than a milling machine which can position the tool automatically.

Table 8.34: Detailed process options for flange component.

feature	op	elem.	Tsetup	Tmachining	machine type	process
itp860605150	0	0	1.253	0.026	lathes	rough_profile_boring
itp860605150	1	0	1.253	0.026	lathes	semi_chamfer_boring
itp860605150	2	0	1.253	0.026	lathes	rough_taper_boring
itp860605028	0	0	1.253	0.026	lathes	rough_profile_boring
itp860605028	1	0	1.253	0.026	lathes	semi_chamfer_boring
itp860605028	2	0	1.253	0.026	lathes	rough_taper_boring
etp860604976	0	0	1.253	0.249	lathes	rough_profile_turning
etp860604976	1	0	1.253	0.373	lathes	semi_chamfering
etp860604976	2	0	1.253	0.249	lathes	rough_taper_turning
etp860604879	0	0	1.253	0.249	lathes	rough_profile_turning
etp860604879	1	0	1.253	0.373	lathes	semi_chamfering
etp860604879	2	0	1.253	0.249	lathes	rough_taper_turning
efa859480299	0	0	1.253	0.511	lathes	rough_facing
efa859480256	0	0	1.253	0.860	lathes	rough_facing
icy859478025	0	0	0.820	0.073	milling_mc	drilling
icy859478025	1	0	0.820	0.266	milling_mc	rough_bore_milling
icy859478025	2	0	1.253	0.073	lathes	drilling
icy859478025	3	0	1.253	0.415	lathes	rough_cyl_boring
icy859478025	4	0	1.253	0.415	lathes	rough_profile_boring
ecy859476095	0	0	1.253	0.270	lathes	rough_cyl_turning
esp859476243	0	0	1.253	0.356	lathes	rough_profile_turning
esp859476243	1	0	1.253	0.348	lathes	rough_cyl_turning
esp859476243	2	0	1.253	1.005	lathes	rough_facing
esp859476923	0	0	1.253	0.331	lathes	rough_profile_turning
esp859476923	1	0	1.253	0.281	lathes	rough_cyl_turning
esp859476923	2	0	1.253	0.860	lathes	rough_facing
bho859478624	0	0	0.820	0.945	milling_mc	drilling
bho859478624	0	1	0.820	1.028	milling_mc	reaming
bho859478624	1	0	0.820	1.028	milling_mc	drilling
bho859478624	2	0	0.820	0.945	milling_mc	drilling
bho859478624	2	1	0.820	0.248	milling_mc	rough_bore_milling
bho859478624	3	0	0.820	1.028	drills	drilling
bho859478624	4	0	0.820	0.945	drills	drilling
bho859478624	4	1	0.820	1.028	drills	reaming
itd860602109	0	0	0.820	9.990	milling_mc	thread_milling
itd860602109	1	0	1.253	0.592	lathes	thread_boring

Two alternative aggregate plans produced by the sequencing and machine selection algorithm are shown in. are shown in Table 8.35. The results show that the sequencing algorithm has successfully reordered the operation elements into a feasible sequence. The first processing is performed on the internal bore, which is first bored to the correct diameter, before the chamfers are applied using a profile boring operation. The thread is then applied using profile boring. The processing then moves to the outside of the cylinder, where the external steps and then the cylinder are cut. The external chamfers are then processed, before the final turning process of facing the component. Once all the axi-symmetric features have been applied, the component is moved to another machine for the twelve holes to be drilled.

Table 8.35: Alternative routes for flange component

	feature	process	machine	*tm*	*tp*	*tt*	cost	sum
	Material						27.0	27.02
0	icy859478025	rough_cyl_boring	Swedturn500	0.41	1.25	0.00	1.58	28.53
1	itp860605028	rough_profile_boring	Swedturn500	0.03	0.00	0.00	0.02	28.56
2	itp860605150	rough_profile_boring	Swedturn500	0.02	0.00	0.00	0.02	28.59
3	itd860602109	thread_boring	Swedturn500	0.59	0.00	0.00	0.65	29.24
4	esp859476923	rough_cyl_turning	Swedturn500	0.60	0.00	0.00	0.66	29.90
5	esp859476243	rough_cyl_turning	Swedturn500	0.52	0.00	0.00	0.57	30.48
6	ecy859476095	rough_cyl_turning	Swedturn500	0.27	0.00	0.00	0.29	30.77
7	etp860604879	rough_profile_turning	Swedturn500	0.37	0.00	0.00	0.41	31.18
8	etp860604976	rough_profile_turning	Swedturn500	0.37	0.00	0.00	0.41	31.60
9	efa859480256	rough_facing	Swedturn500	1.71	0.00	0.00	1.89	33.49
10	efa859480299	rough_facing	Swedturn500	0.34	0.00	0.00	0.37	33.86
11	bho859478624	drilling	Cincinnati25HC	1.02	0.82	0.01	1.86	35.73
	Material						27.0	27.02
0	icy859478025	rough_cyl_boring	Swedturn12x4	0.41	1.25	0.00	1.58	28.53
1	itp860605028	rough_profile_boring	Swedturn12x4	0.02	0.00	0.00	0.02	28.56
2	itp860605150	rough_profile_boring	Swedturn500	0.02	1.25	0.00	1.15	29.72
3	itd860602109	thread_boring	Swedturn500	0.59	0.00	0.00	0.65	30.37
4	esp859476923	rough_cyl_turning	Swedturn500	0.60	0.00	0.00	0.66	31.03
5	esp859476243	rough_cyl_turning	Swedturn500	0.52	0.00	0.00	0.57	31.61
6	ecy859476095	rough_cyl_turning	Swedturn500	0.27	0.00	0.00	0.29	31.90
7	etp860604879	rough_profile_turning	Swedturn500	0.37	0.00	0.00	0.41	32.31
8	etp860604976	rough_profile_turning	Swedturn500	0.37	0.00	0.00	0.41	32.72
9	efa859480256	rough_facing	Swedturn500	1.71	0.00	0.00	1.89	34.62
10	efa859480299	rough_facing	Swedturn500	0.34	0.00	0.00	0.37	34.99
11	bho859478624	drilling	CincinnatiT10	1.02	0.82	0.02	1.87	36.86

The first of the routes is the best produced by the routing algorithm. A single lathe is used for the first ten stages of processing, with only the one necessary machine tool change to switch to the drill at the end. The total cost of the component is £35.70 using this route. The alternative route is sub-optimal, since it uses two lathes and therefore the cost is increased to £36.90. Although there are ten operation elements scheduled for the same machine tool, there is no shortage of tool positions since many elements use the same process as others.

8.6 Testing overall system performance

This section addresses the issues of the success of the methodology, as distinct from the computer system. The key question to be answered here is will CESS provide a useful tool to product development, bringing real benefits? The most powerful features of CESS are the provision of an expert knowledge sources for manufacturing planning, encompassing a variety of manufacturing options for each product design and the provision of an automated system for applying process knowledge in order to rapidly evaluate designs. It is expected that product designers would benefit from both of these features, since they will be empowered with the ability to bring processing knowledge to bear on the early designs. Production planning engineers can be expected to use the second of the features, however. Already possessing process knowledge, they will gain by the ability to perform assessments more rapidly. In particular, CESS gives the ability to consider multiple processes and to investigate the effects on production costs of a range of product development decisions, including factory layout and equipment changes as well as design changes.

8.6.1 Process identification

It can be seen from the previous examples that CESS can successfully identify manufacturing processes for individual features. An important aspect of this testing was to verify that the constraint functionality of the processes was operating correctly: Each process has certain quality constraints which may dictate the use of multiple processing steps, or preclude the use of the process in a particular instance. Constraints can be both due to high and low quality settings, the former would preclude the use of the process for finishing the feature, whilst the latter would imply the need for a roughing process. In addition, constraints are set on processes which require certain geometries in order to be used. Key examples of this are the hole finishing processes such as reaming and boring. These processes can only be used if the rough hole has been created (or is provided by another feature). The system is designed to force the planning of jobs to create such access holes where possible. The testing process has established that this functionality was operating as intended.

For ease of use, the results from this function must be passed to the user in a clear manner. Figure 8.10 shows an example of the output of the process option generation stage. For each feature a breakdown of the process alternatives is given, within the number of processing steps for each option clearly visible. Further development of this interface would include the process names for each operation step; at present, this information is available by clicking on the option with the cursor.

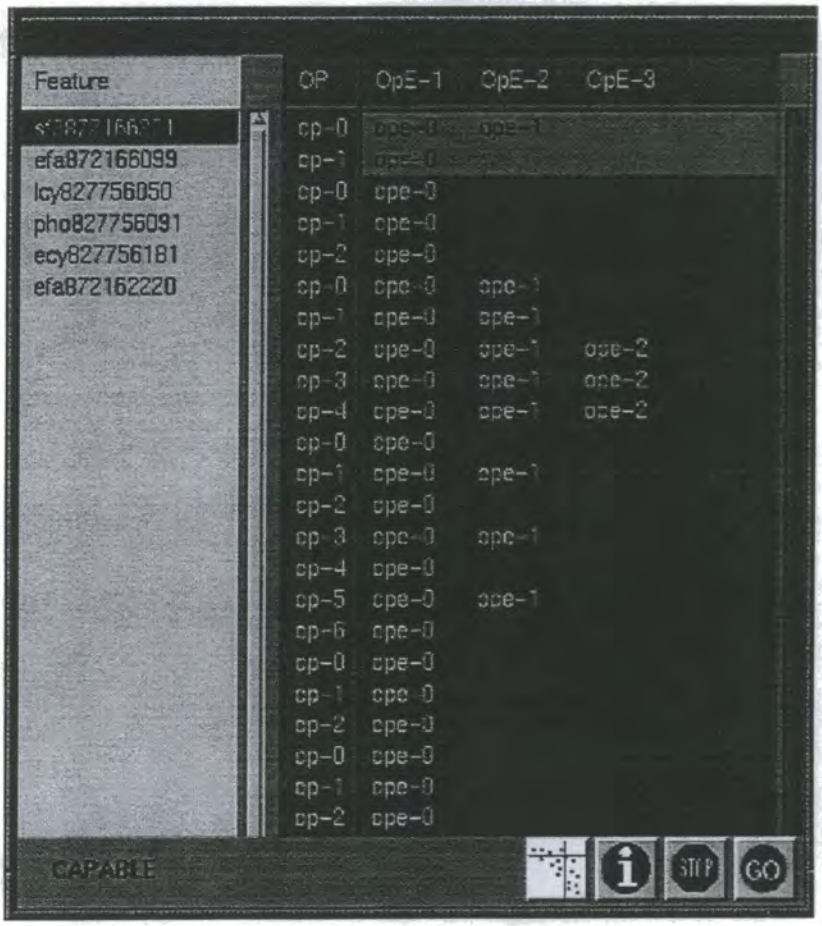


Figure 8.10: Process options window

8.6.2 Process (and machine tool) evaluation

As described in previous chapters, the evaluation of individual process and machine tool selections is performed by process specific methods which have been developed to calculate manufacturing time and by a generic method to estimate production quality. This evaluation is carried out on an individual level for each operation element (at process stage) or each machine option (at machine tool stage). The times and quality levels calculated are then used by the next stage of the planning process to select from the alternative options. The difference between the process and machine tool evaluation functions is that for process evaluation the machine tool data required by the different algorithms is calculated as the average of the machine tools available in a class, instead of using the particular parameters of a selected machine tool.

The results of these evaluations are communicated to the user in a number of ways: each operation listed on the process option screen can be queried for a detailed cost breakdown, which shows the operation elements which are required, the machine tool type (or specific machine tool) for each and the portion of material removal which the elements perform, in addition to relevant cost, quality and time data. Figure 8.12 shows the operation detail windows for the process and machine selection stages. The user is able to examine individual process steps to determine planning material removal and to review predicted times, cost and quality. In addition, a breakdown of the results is written to an output file for reference.

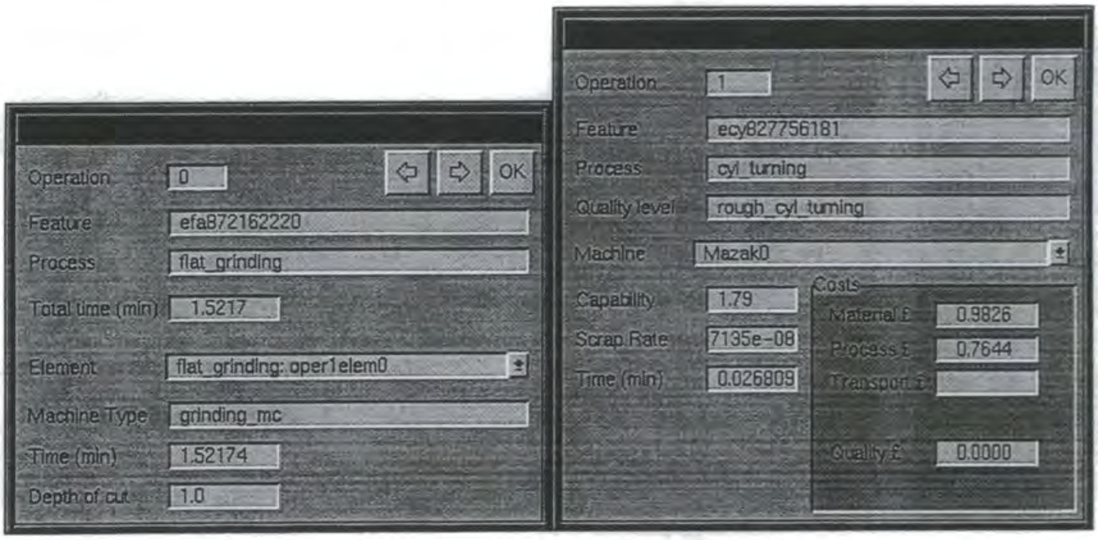


Figure 8.11: Operation detail windows, process and machine selection

8.6.3 Selection method testing

The assessment of the selection functions of the system was divided into a number of strategies. The effectiveness of the genetic algorithm in finding optimal routes was shown by the use of systems with known best routings, such as occasions where a single machine tool could be used for all operations, thus eliminating set-up times. The GA technique was shown to be capable of identifying such routes within very few generations. Additionally, it is possible to monitor the progress of the GA in reducing the objective function (component cost) over many generations. In testing, the number of generations used and the size of the population were varied greatly to find the most suitable values. Those selected were chosen as the best compromise between optimising selection and rapid analysis.

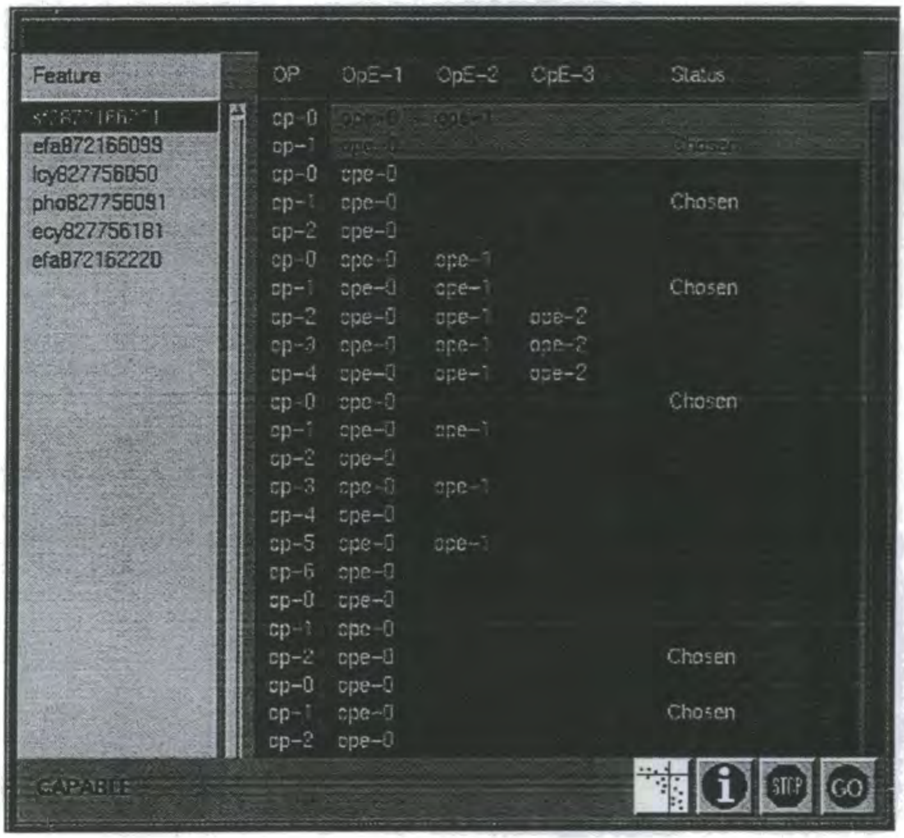


Figure 8.12: Process options window with selected operations

Figure 8.12 show the results of the selection process as displayed to the user in the process options window. More detailed information can be obtained from this by clicking on individual items, which causes the process details window to be opened.

8.6.4 Sequence algorithm tests

To test the sequence generation algorithm, it was necessary to run the full aggregate process planning system and look at the resulting sequences. To generate more sequences from the same model set, deliberately sub-optimal process selections were made, which introduced a greater variation of processes and machines into the sequencing problem. The principle problems for the sequencing algorithm lie in features which have precedence relationships such as holes and threads. The testing of the sequence algorithm which has been undertaken has established that the basic system works when there are few additional constraints due to quality relationships. Further testing is required to prove the method for more complex cases where it is expected that the will be some cases where constraints conflict and require human intervention to determine the best solution.

8.7 Conclusions

The testing and evaluation of the proposed methodology has been undertaken through the use of CESS. It has been demonstrated that CESS is capable of generating aggregate process plans from aggregate product model data, and that these plans are feasible. The times and costs calculated by the system are broadly comparable with times observed in industry. A more thorough testing of the system would require access to more detailed cost breakdowns from industry than were available during this project. In particular, the testing of the system is sensitive to the costing methods applied. However, the times calculated for processing have been shown to be realistic estimates for production times, therefore it is assumed that the cost calculations are also valid.

It was unfortunately not possible to test the system in a working design environment where real-time design changes could be rapidly assessed and therefore no firm conclusions can be drawn as to the feasibility of such a system as part of an integrated concurrent engineering system. Follow on work is under-way to provide testing of the concepts outlined in this thesis in an industrial setting in order to validate this aspect of the methodology.

Chapter 9

Discussion and conclusions

9.1 Discussion

A method for the assessment of manufacturing options of early product designs has been developed. This has been implemented as a computer system, CESS, which generates aggregate process plans from a product model, and has been implemented on a UNIX platform using *Smart Elements* 3.1 for X-Windows. The system maintains models of the product design, the manufacturing facility and production processes. The various functions of the system are integrated using a graphical user interface to provide a system which is flexible and easy to use. One of the important considerations in the design of the system was in structuring it so that it could be used in different ways depending on the data available and the user's expertise.

A comprehensive review of published literature was conducted covering computer aided engineering, product modelling, process modelling and the disciplines of product development, including design and process planning. The adoption of concurrent engineering as a product development strategy leads to a requirement for a restructuring of all design and manufacturing disciplines. In particular, the computer tools which are available for supporting design and process planning need to be integrated at an earlier stage than is presently possible if the ideal of concurrent working on process plans and design is to be realised. Previous attempts to achieve this integration met with varying success. Some systems attempt to increase the automation of the link between "traditional" CAD and CAPP systems but do not account for the lack of detail available in designs. Other systems concentrate on main process selection but do not provide enough detail in process plans, leaving no room for refinement with increasing design detail or for consideration of the available resources. Some concurrent engineering systems have been designed to use a completely new design environment, instead of the CAD approach. Such systems may result in models which are more easily translated into process plans, but may hamper the design process, as they require more specific inputs from the user.

Aggregate process planning has been identified as the most suitable strategy for overcoming these problems. This approach recognises that for a given level of design detail, there are certain process planning functions which can be performed and there are some which cannot. In addition, the CESS implementation has the ability to make process assessments on a feature by feature basis, so that the gradually increasing detail of the product model may be assessed. In performing such analysis, it is recognised that additional detailing of products will inevitably result in an increase in predicted cost. Careful management of the way in which the aggregate process planning function is applied will be required to ensure that it is made clear which cost changes are the result of changes to previously costed design elements and which are the result of additional refinements to the design details. The ability to regularly update the estimated cost of the design as the detail is added should give designers a better understanding of the sources of costs within the product and encourage simpler, more efficient designs.

A number of requirements have been identified for supporting concurrent engineering. Designers would benefit from a ready review of potential processing options which are available for their latest design. The ability to get detailed feedback on the manufacturing consequences of design modifications would encourage the consideration of alternative designs during the conceptual and embodiment stages. Production facility designers need to be made aware of the requirements which a new product design will place on the existing factory, and so manufacturing assessment of product designs needs to include a link to the machine tools and other resources required.

The data which is used in CESS has been gathered from public domain sources. The process models are based on standard cutting parameters which have been determined by taking the mean values of tool manufacturer's data. Tool related process constraints have similarly been based on the full range of tool catalogues. For use by a particular company, it would be possible to modify these standard cutting data to reflect the policy of the company and to set tool sizes to the limits of tools actually used in the factory. The process quality levels which are used have been drawn from engineering textbooks. Again it would be possible to tailor these levels to suit the known capabilities of the processes as used by the company, with the number of distinct quality levels applied to a process being modified to reflect the way a company applies the processes. For example, some companies only recognise two forms of turning: roughing and finishing, discarding the concept of semi-finishing. Finally, quality data for the prediction of scrap rates relies on actual data retrieved from SPC systems and therefore gives a direct link between CESS and the factory performance.

One of the main functionalities involved in aggregate process planning is in determining the best combination of process and machine options for multiple features. The CESS implementation uses genetic algorithms to find the ideal combinations. The use of genetic algorithms in process planning is not a new concept, having been successfully used by other researchers. The genetic algorithm approach allows the optimisation to be performed at feature level and gives the added benefit of automatically identifying valid alternative solutions. CESS, therefore, has an advantage over other purely knowledge based process planning systems in that the effects of one feature on the overall process plan may be recognised. Whilst some systems are forced to suggest the same process options for all features of a type, the CESS aggregate process plans can suggest alternative process routes for two features of the same type if quality specifications or feature size require it, even during the early design stages.

The complexity of the methodology developed in this thesis is such that it could only be tested by implementing the algorithms as a computer system. A software language which combined a knowledge based system approach was required to build and manage models of process planning expertise, whilst the need to maintain a feature based product model led to a requirement for an object oriented language. The *Smart Elements* software development package was selected, as described in Chapter 3.5. A particular benefit of this system was that it is designed as a rapid prototyping tool for software systems and is therefore eminently suited to the development of research prototypes. In these systems the goal of the development is not to produce a fully functioning and stable system, but to identify and resolve difficulties in the methodology and to perform the necessary calculations to test it. It is worth noting, however, that a substantial part of the development time for this project was spent on building a suitable interface to the system so that the aggregate process planning methodology could be demonstrated in the proposed environment of an integrated system.

9.2 Conclusions

The research described in this thesis has addressed the following issues:

- There is a recognised need for a closer link between design decisions and production consequences. This can best be achieved by empowering the design engineer with the ability to assess the manufacturing options available for a design.

- The concurrent engineering methodology requires that the process planning function be initiated earlier in the design cycle, at a stage when less information is available about the design. Conventional CAPP systems are unsuited to use without fully detailed design data.

- It is necessary to compare and select between alternative process options at an early stage if designs are to be optimised for a particular process.

- Human process planners are not able to evaluate the large numbers of alternative options available for production of complex products, leading to a selection based on intuition instead of calculation. In addition, the rapidly changing nature of production facilities means there is a need for immediate availability of alternative production plans.

In order to address these problems:

- A methodology for aggregate process planning has been developed which incorporates an automated process planning system at the aggregate level and the assessment of production costs and times using detailed process models which can operate on reduced product data.

- A product model has been developed which allows the representation of machinable components, including the specification of quality requirements.

- A generic process modelling technique has been developed and applied to selected machining processes to develop methods for calculation of production criteria including cost and time.

- A resource model has been developed for the representation of manufacturing facilities at the factory, cell and machine tool levels.

- A generic machining quality assessment methodology has been developed.

- The above systems have been implemented on a UNIX based computer. Testing of the software has yielded encouraging results.

The research described in this thesis demonstrates novelty in the following ways:

- The application of detailed process models to incomplete design data in order to assess the manufacture of designs. Previous attempts to apply comparative models of process options to designs for process selection have used over-simplified product models which do not analyse the requirements at the level of individual process steps. This new approach enables the system to take into account individual factory circumstances such as the presence of specialist machine tools.

- The minimum information requirements for aggregate process planning during early design have been identified and a product model encompassing this data has been developed.

- In contrast to most process selection methodologies, a direct link is made between the developing product model and the production facility, via captured knowledge of production processes.

- As the system generates sets of alternative process plans, the process planner may select the most suitable route for detailed process planning depending on the latest factory conditions. The alternative aggregate plans could be used as an input into a distributed shop-floor process planning system.

- The system provides a flexible environment in which the same functionality may be used to assess the inter-connected effects of changes to designs, process plans and facilities.

- The quality prediction module of CESS provides a means of combining standard quality ranges for processes and machine tools with measured data from the shop floor. Whereas most quality systems are designed to indicate whether or not a process is suitable for a process, the CESS system aims to predict process capabilities and to cost the consequences of a particular capability for a given product.

9.2.1 Recommendations for further work

This work has led to the identification of many further avenues for research and development. In this section a number of possible extensions to the work are discussed. Many of the research areas identified during the course of this project have already begun to be researched and the resulting computer system is undergoing further development as part of an EPSRC funded research project.

The proposed methodology of product development using aggregate process planning requires that the process planning system has the capability of assessing all feasible production processes for a component. The current implementation of CESS covers only a sub-set of machining processes, due to the limitations of time. Further work is required to enhance the process model and to expand it to include additional processes. In particular, the current system has no models for sheet-metal working processes or for chemical and electro-chemical processes.

The resource model of CESS is at present underdeveloped and there are opportunities for improving this model in several ways. The class structure developed for machine tools could be refined, both by increasing the number of classes to reflect the subtle variations in machine capabilities, and by adding further detail to the models for each machine type. The machines model described in this thesis was primarily developed to test the process planning rules. It is not intended to be a fully comprehensive and definitive list of all machine types.

The aggregate process planning methodology could be extended to include joining processes, including assembly and welding. This would enhance the power of the methodology since it would increase the range of design configurations which could be compared using the system. It is thought that an aggregate process planning system including assembly modelling would prove an improvement over the design for assembly (DFA) methodology, since the system would be able to compare the cost of alternative configurations, including both machining and assembly costs. This approach would not suffer from the problem faced by DFA of reducing assembly costs only to cause a larger increase in machining costs.

Whilst this thesis has concentrated on the main uses of CESS as a system for rough-cut process planning and evaluating the manufacturing cost of a design, a number of additional modes of operation would be possible with some extra development. In particular, the ability to model multiple manufacturing facilities using a generic method should allow the system to be used as part of a facility design system. By varying the facility model whilst retaining the same products, the effect of alternative layouts or additional equipment could be determined. An extension of this work would be the development of an automated system for facility design based on the CESS output of aggregate process plans. Some work has been conducted in this area, with the development of a clustering algorithm designed to operate on CESS output (Baker and Maropoulos, 1997). An alternative use of the aggregate process planning methodology is as a benchmarking system to compare the performance of a factory with a state-of-the-art factory model.

CESS was developed as a test-bed for the aggregate process planning methodology and to prove the concept of using aggregate process plans to evaluate early product designs.

It was not designed to be applied in the field and therefore several functions which would be required to turn it into a fully functioning system are not in place. In particular, there is a need for a link to a commercial CAD package on which the designs would be developed. This should be accomplished using the STEP standard. However, a requirement for this link is for a means of extracting from a detailed product model only that information which is required by the aggregate product model. During this project, no solid model based CAD system was available so this function was not attempted. The quality assessment function of CESS is another element which has a requirement for additional functionality. The assessment is based on pre-processed data from the factory SPC system. The data used in testing the system was gathered and entered by hand from the shop-floor. A commercial application of a system like this would have a requirement for an automated method of retrieving the SPC data and formatting them for use by CESS.

References

Abdalla, HS, Goulden, CR, and Pancholi, H (1995), 'A Knowledge-Based Process Quality-Control Planning System For Design For Manufacturability', 11th national conference on manufacturing research, De Montfort University, Leicester, UK, 1995, published in *Advances In Manufacturing Technology IX*, 467-471.

Abdalla, HS, and Knight, J (1994), 'An expert-system for concurrent product and process design of mechanical parts', *Proceedings of the Institute of Mechanical Engineers part B-journal of engineering manufacture*, vol. 208, no. 3, 167-172.

Ajmal, A, and Zhang, HG (1994), 'State-of-the-art review of CAPP: Its impact and its applications in CIM', 10th national conference on manufacturing research, Loughborough, 5th-7th September, 1994, published in *Advances in Manufacturing VIII.* (K Case and ST Newman eds.), 239-243.

Akao, Y (1990), 'Quality Function Deployment: Integrating Customer Requirements into Product Design', Productivity Press.

Allen, AJ, and Swift, KG (1990), 'Manufacturing process selection and costing', *Proceedings of the Institute of Mechanical Engineers part b-journal of engineering manufacture*, vol. 204, no. 2, 143-148.

An, D, Leep, HR, Parsaei, HR, and Nyaluke, AP (1995), 'A product data exchange integration structure using PDES/STEP for automated manufacturing applications', *Computers and Industrial Engineering*, vol. 29, no. 1-4, 711-715.

Andreason, MM (1991), 'Design for production - overview of methodology', *International conference on Engineering Design, ICED '91.*

Andreason, MM (1991), 'Design methodology', *Seminar on Design Theory and its Application, The Norwegian academy of technical sciences.*

Andreason, MM, and Gudnason, CH (1992), 'Planning and Integration of Product Development', Gavrial Salvendy (ed.), *Handbook of industrial engineering, 2nd edition*, John Wiley & sons.

Andreason, MM, and Oleson, J (1990), 'Strengthening the DFA rules by dispositional mechanisms', *Proceedings of the 11th Conference on Assembly Automation.*

Andreason, MM, and Oleson, J (1990), 'The concept of dispositions', *Journal of Engineering Design*, vol. 1, no. 1, 17-36.

Arai, E, and Iwata, K (1992), 'Product modeling system in conceptual design of mechanical products', *Robotics and computer-integrated manufacturing*, vol. 9, no. 4-5, 327-334.

Arciszewski, T, and Karni, R (1997), 'Ideation-based conceptual design of a materials handling and transfer (MHT) system', *Proceedings of the 32nd International MATADOR Conference*, 35-40.

Baker, RP, and Maropoulos, PG (1997), 'Manufacturing capability assessment for cellular manufacturing systems', accepted for publication by *International journal of production research.*

Batchelor, R, and Swift, KG (1996), 'Conformability analysis in support of design for quality', *Proceedings of the Institute of Mechanical Engineers part b-journal of engineering manufacture*, vol. 210, no. 1, 37-47.

Betteridge, M, and Maropoulos, PG (1996), 'A paper about mike's work', 12th National Conference on Manufacturing Research, Bath University, Published in *Advances in Manufacturing Technology X*

Birolini, A (1993), 'Design for reliability', *Concurrent engineering: automation, tools and techniques*, 307-348.

Boothroyd, G, and Alting, L (1992), 'Design for assembly and disassembly', *Annals of the CIRP*, vol. 41, no. 2, 625-636.

Boothroyd, G, and Dewhurst, P (1987), 'Product design for assembly', Boothroyd-Dewhurst Press, USA.

Brandon, JA, and Huang, GQ (1993), 'Use of an agent-based system for concurrent mechanical design', *Concurrent engineering: automation, tools and techniques*, 463-479.

Burgess, AG, Morgan, I, and Vollmann, TE (1993), 'Cellular manufacturing: its impact on the total factory', *International journal of production research*, vol. 31, no. 9, 2059-2077.

Butterfield, WR, Green, MK, Scott, DC and Stoker, WJ (1985), 'Part features for process planning', *C-85-PPP-03, CAMI-I Inc.*, Arlington, Texas, USA..

Carpenter, ID (1996), 'Machinability assessment and tool selection for milling', PhD Thesis, University of Durham.

Cartwright, AJ (1997), 'Interactive prototyping - a challenge for computer based design', *Research in engineering design-theory applications and concurrent engineering*, vol. 9, no. 1, 10-19.

Case, K (1994), 'Using a design by features cad-system for process capability modelling', *Computer integrated manufacturing systems*, vol. 7, no. 1, 39-49.

Case, K, and Gao, J (1993), 'Feature technology - an overview', *International journal of computer integrated manufacturing*, vol. 6, no. 1-2, 2-12.

Case, K, Gao, JX, and Gindy, NNZ (1994), 'The implementation of a feature-based component representation for cad-cam integration', *Proceedings of the Institute of Mechanical Engineers part b-journal of engineering manufacture*, vol. 208, no. 1, 71-80.

Chan, FTS, Jayaprakash, B, and Tang, NKH (1995), 'Design of an automated cellular manufacturing system with simulation modelling: a case study', *International journal of computer applications in technology*, vol. 8, no. 1, 1-11.

Chu, XN, and Holm, H (1994), 'Product manufacturability control for concurrent engineering', *Computers in industry*, vol. 24, no. 1, 29-38.

Clarke, C (1997), 'Strand simplicity, product review', *CADCAM magazine*, May issue, 33.

Colton, JS (1993), 'An intelligent design for manufacture system', A.Kusiak (ed.), *Concurrent engineering: automation, tools and techniques*, John Wiley and sons, 153-176.

Davies, BJ, Joseph, A, and Kalta, M (1991), 'An expert system to generate NC data from CAD product models', *Journal of materials processing technology*, vol. 26, no. 2, 227-236.

Darlington, J, and Maropoulos, PG, (1997), 'Supply chain modelling for responsive manufacture', 13th national conference of manufacturing research, Glasgow Caledonian University, published in *Advances in manufacturing XI* (DK Harrison, ed.), p98-102.

Dixon, JR (1988), 'Designing with features: building manufacturing knowledge into more intelligent CAD systems', *ASME Manufacturing International '88*, Atlanta, Georgia, USA, 51-57.

Elmaraghy, WH (1993), 'Evolution and Future perspectives of CAPP', *Annals of the CIRP*, vol. 42, no. 2, 739-751.

Eppinger, SD, Whitney, DE, Smith, RP, and Gebala, DA (1994), 'A model-based method for organizing tasks in product development', *Research In Engineering Design-Theory Applications And Concurrent Engineering*, vol. 6, no. 1, 1-13.

van't Erve, AH and Kals, HJJ, (1988), 'XPLANE, a generative computer aided process planning system for part manufacturing', *Annals of the CIRP*, vol 35, no. 1, 325-329.

Esawi, AMK (1994), 'Systematic process selection in mechanical design', PhD. Thesis, University of Cambridge, England.

ESPRIT 6805 (1995), 'COncurrent Manufacturing PLANning and shop control for small batch production', A.I. Systems Brussels.

Farmer, LE, and Gladman, CA (1986), 'Tolerance technology - computer-based analysis", *Annals of the CIRP,* vol 35, no. 1, 7-10.

Fu, Z, de Pennington, A, and Saia, A (1993), 'A graph grammar approach to feature representation and transformation', *International journal of computer integrated manufacturing*, vol. 6, no. 1-2, 137-151.

Gandhi, A, and Myklebust, A (1989), 'A natural language approach to feature based modelling', Mobil research and development corporation, Princeton, New Jersey, USA.

Gershenson, J, and Ishii, K (1993), 'Life-cycle serviceability design', A.Kusiak (ed.), *Concurrent engineering: automation, tools and techniques*, John Wiley and sons.

Gindy, NNZ (1989), 'A hierarchical structure for form features', *International journal of production research*, vol. 27, no. 12, 2089-2103.

Gindy, NNZ, Huang, X, and Ratchev, TM (1993), 'Feature-based component model for computer-aided process planning systems', *International journal of computer integrated manufacturing*, vol. 6, no. 1-2, 20-26.

Gindy, NNZ, Ratchev, T, and Case, K (1994), 'Process capability models for equipment selection', 10th National Conference on Manufacturing Research, Loughbough University of Technology, September 5th-7th 1994, published in *Advances In Manufacturing Technology VIII* (K Case and ST Newman, eds.), 244-248.

Hartley, JR (1990), *Concurrent engineering, shortening lead times, raising quality and lowering costs*, Productivity press.

Haudrum, H, and Alting, L (1993), 'Structured methods for process selection in the stage of design', *ICCIM, Singapore, Sept.*

He, JR (1991), 'Tolerancing for manufacturing via cost minimization', *International Journal Of Machine Tool Design And Research*, vol. 31, no. 4, 455-470.

Henderson, MR (1986), 'Automated group technology coding from a three-dimensional CAD database', S.C.Lu, and R.Komanduri (ed.), *Knowledge-based expert system for manufacturing*, ASME, 195-204.

Herman, A, Lawley, and Lu (1993), 'An opportunistic approach to process planning within concurrent engineering environment', *Annals Of The CIRP*, vol. 41, no. 1.

Ishii, K (1993), 'Modelling of concurrent engineering design', A.Kusiak (ed.), *Concurrent engineering: automation, tools and techniques*, John Wiley and sons, 19-40.

ISO (1994), 'Standard for the exchange of product data .1. Introduction'.

ISO (1994), 'Standard for the exchange of product data .11. Express reference manual'.

Jayaraman, R and Srinivasan, V (1989), 'Geometric tolerancing: 1. Virtual boundary requirements', *IBM journal of research and development*, vol 33, no. 2, 90-104.

Kandiller (1994), 'A comparative study of cell formation in cellular manufacturing systems', *International journal of production research*, vol. 32, no. 10, 2395-2429.

Kannapan, SM, and Marshek, KM (1992), 'A schema for negotiation between intelligent design agents in concurrent engineering', *Ifip Transactions B-Applications In Technology*, vol. 4, 1-20.

Krause, F-L, Kimura, F, Kjellburg, T, and Lu, SC-Y (1993), 'Product Modelling', *Annals of the CIRP*, vol. 42, no. 2, 695-706.

Krause, F-L, Ulbrich, A, and Woll, R (1993), 'Methods for quality-driven product development', *Annals Of The CIRP*, vol. 42, no. 1, 151-154.

Krause, FL, and Ochs, B (1992), 'Potentials of advanced concurrent engineering methods', *Ifip Transactions B-Applications In Technology*, vol. 6, 15-28.

Kristensen, L, and Lenau, T (1993), 'MADED - an information system for manufacturing processes', *Int. Conference On Engineering Design, ICED 93*, 1093-1096.

Kruth, JP (1992), 'New manufacturing techniques for rapid prototyping and concurrent engineering', *Ifip Transactions B-Applications In Technology*, vol. 6, 51-82.

Kulkarni, VS, and Pande, SS (1996), 'Representation of feature relationship tolerances in solid models', *International journal of production research*, vol. 34, no. 7, 1975-1994.

Kunz, JC, Luiten, GT, Fischer, MA, Jin, Y, and Levitt, RE (1996), 'CE4 - concurrent engineering of product, process, facility, and organization', *Concurrent engineering-research and applications*, vol. 4, no. 2, 187-198.

Kusiak, A, and Wang, J (1993), 'Decomposition in concurrent design', Kusiak, A (ed.), *Concurrent engineering: automation, tools and techniques*, John Wiley & Sons, Inc, 481-507.

Kusiak, A, and Wang, J (1993), 'Decomposition of the design process', *Journal Of Mechanical Design*, vol. 115, no. 4, 687-695.

Latif, MN, Boyd, RD, and Hannam, RG (1993), 'Integrating cad and manufacturing intelligence through features and objects', *International journal of computer integrated manufacturing*, vol. 6, no. 1-2, 87-93.

Lenau, T, and Alting, L (1992), 'Models of manufacturing processes for design information systems', *Manufacturing Systems*, vol. 21, no. 2, 129-135.

Lenau, T, and Mu, L (1993), 'Features in integrated modeling of products and their production', *International journal of computer integrated manufacturing*, vol. 6, no. 1-2, 65-73.

Liu, MC (1993), 'An investigation of multivariate process capability index', *2nd Industrial Engineering Research Conference Proceedings*, Wichita State University, Dept. of Industrial Engineering, Wichita, KS, USA, 67260, 644-648.

Lu, SC-Y (1990), 'Knowledge processing for concurrent engineering: and evolving challenge in cim research', *Robotics And Computer Integrated Manufacturing*, vol. 7, no. 3/4, 263-277.

Maropoulos, PG (1995a), 'A novel process planning architecture for product based manufacture', *Proceedings Of The Institute of Mechanical Engineers Part B*, vol. 209, 267-275.

Maropoulos, PG (1995b), 'Review of research in tooling technology, process modelling and process planning .1. tooling and process modelling', *Computer integrated manufacturing systems*, vol. 8, no. 1, 5-12.

Maropoulos, PG (1995c), 'Review of research in tooling technology, process modelling and process planning .2. process planning', *Computer integrated manufacturing systems*, vol. 8, no. 1, 13-20.

Maropoulos, PG (1995d), 'Aggregate process modelling technology for process planning applications', *Proceedings of the 11th International conference on Computer-Aided Production Engineering*, I.Mech.E, London, 179-184.

Medland, AJ, and Mullineux, G (1993), 'A constraint approach to feature-based design', *International journal of computer integrated manufacturing*, vol. 6, no. 1-2, 34-38.

Meerkamm, H (1993), 'Design system mfk - an important step towards an engineering workbench', *Proceedings of the Institute of Mechanical Engineers part b-journal of engineering manufacture*, vol. 207, no. 2, 105-116.

Mei, J, and Zhang, HC (1992), 'Tolerance analysis for automated setup selection in CAPP', *Production Engineering and Design - Concurrent Engineering, Transactions of the ASME*, vol. 59, 211-228.

Mezgar, I, and Kovacs, G (1993), 'CE approach to quality assurance of products and production systems', *Ifip Transactions B-Applications In Technology*, vol. 13, 259-266.

Mittal, RO, Irani, SA, and Lehtihet, EA (1990), 'Tolerance control in the machining of discrete components', *Journal Of Manufacturing Systems*, vol. 9, no. 3, 233-246.

Oberg, E (1992), 'Machinery's handbook, 24th edition', Industrial Press Inc., New York, USA.

Pahl, G, and Beitz, W (1984), 'Engineering design', *The Design Council*, Springer Verlag, London, England.

Panchal, K, Raman, S, and Pulat, PS (1992), 'Computer-aided tolerance assignment procedure (CATAP) for design dimensioning', *International Journal Of Production Research*, vol. 30, no. 3, 599-610.

Perzyk, M, and Meftah, OK (1995), 'Selection of manufacturing process in mechanical design', J.A.McGeough, and S.Strzelczak (ed.), *Proceedings of the 13th International conference on Computer-Aided Production Engineering*, Warsaw, Poland, 243-248.

Peters, J, Krause, KL, and Agerman, E (1990), 'Design: an integrated approach', *Annals Of The CIRP*, vol. 39, no. 2, 599-607.

PHOENICS (1992), 'The PHOENICS beginners manual'. *CHAM TR/100*, CHAM, Bakery House, Wimbledon, London, England.

Ping, J (1994), 'An automatic tolerance assignment approach for tolerance charting', *International journal of advanced manufacturing technology*, vol. 9, 362-368.

Pratt, MJ (1993), 'Applications of feature recognition in the product life-cycle', *International journal of computer integrated manufacturing*, vol. 6, no. 1-2, 13-19.

Pratt, MJ, and Wilson, PR (1985), 'Requirements for support of form features in a solid modelling system', *T-85-ASPP-01, CAM-I Inc.*, Arlington, Texas, USA.

Qing, S, and Ming, X (1997), 'A computer-aided modeling, programming and control environment for manufacturing automation', *Proceedings of the 13th International conference on Computer-Aided Production Engineering*, 165-172.

Requicha, AAG, and Chan, SC (1986), 'Representation of geometric features, tolerances, and attributes in solid modelers based on constructive geometry', *IEEE Journal of Robotics and Automation*, vol. RA-2, no. 3, 156-166.

Salomons, OW, Van Houten, FJAM, and Kals, HJJ (1992), 'Review of research in feature-based design', *Journal Of Manufacturing Systems*, vol. 12, no. 2, 114-132.

Sandvik Cormant (1996) *Rotating tools*, C-1100:5-ENG, Sandvik Cormant, Sweden.

SECO Tools AB (1996) *Turning main catalogue*, ST944739E, Sweden.

Shah, JJ, and Rogers, MT (1988), 'Functional requirements and conceptual design of the feature-based modelling system', *Computer-Aided Engineering Journal*, February, 9-15.

Sohlenius, G (1992), 'Concurrent Engineering', *Annals of the CIRP*, vol. 41, no. 2, 645-655.

Spur, G, Krause, F-L, and Armbrust, P (1986), 'Product Models as Basis for Integrated Design and Manufacturing', *Int. J. Mach. Tool. Des. Res.*, vol. 26, no. 2, 171-178.

Su, D, and Wakelam, M (1997), 'Intelligent hybrid system for integration in design and manufacture', J.A.McGeough, and S.Strzelczak (ed.), *Proceedings of the 13th International conference on Computer-Aided Production Engineering*, Warsaw, 265-272.

Swift, KG, and Allen, AJ (1994), 'Product variability risks and robust design', *Proceedings of the Institute of Mechanical Engineers part b-journal of engineering manufacture*, vol. 208, no. 1, 6-19.

Taguchi, G, Elsayed, EA, and Hsiang, T (1989), 'Quality in production systems', McGraw-Hill, London, England.

Thornton, AC (1993), 'Constraint specification and satisfaction in embodiment design', PhD Thesis, Cambridge University, Cambridge, England.

Tonschoff, HK, Aurich, JC, Ehrmann, M, and D'Agostino, N (1996), 'A unified approach to free-form and regular feature modelling', *Annals of the CIRP*, vol. 45, no. 1, 125-128.

Toye, G, Cutkosky, MR, Leifer, LJ, Tenenbaum, JM, and Glicksman, J (1994), 'Share - a methodology and environment for collaborative product development', *International Journal Of Intelligent & Cooperative Information Systems*, vol. 3, no. 2, 129-153.

Turner, JU (1993), 'A feasibility space approach for automated tolerancing', *Journal Of Engineering For Industry-Transactions Of The Asme*, vol. 115, no. 3, 341-346.

van der Net, AJ, de Vries, WAH, Delbressine, FLM, and van der Wolf, ACH (1996), 'A relation based product model suited for integrating design and manufacturing', *Annals of the CIRP*, vol. 45, no. 1, 161-164

van Houten, FJAM, van't Erve, AH, Boogert, RM, Nauta, JM, and Kals, HJJ (1989), 'PART, a feature based computer aided process planning system', *Proceedings of the 21st CIRP International Seminar on Manufacturing Systems*, Stockholm, Sweden, 699-710.

Vancza, J, and Markus, A (1993), 'Features and the principle of locality in process planning', *International journal of computer integrated manufacturing*, vol. 6, no. 1-2, 126-136.

Wan, AR, Harun, W, and Case, K (1994), 'Object-oriented feature-based design', 10th National Conference on Manufacturing Research, Loughbough University of Technology, September 5th-7th 1994, published in *Advances in Manufacturing Technology VIII* (K Case and ST Newman, eds.), 294-298.

Wong, TN, and Leung, CB (1995), 'Feature conversion between neutral features and application features', *Computers & industrial engineering*, vol. 29, no. 1-4, 625-629.

Woodford, CH, Passaris, EKS, and Bull, JW (1992), *Engineering analysis using PAFEC finite element software*, Blackie, Glasgow, Scotland.

Yoshikawa, H (1993), 'Systematization of design knowledge', *Annals Of The CIRP*, vol. 42, no. 1, 131-134.

Zhang, G (1996), 'Simultaneous tolerancing for design and manufacturing', *International journal of production research*, vol. 34, no. 12, 3361-3382.

Zhang, HC, and Huq, ME (1992), 'Tolerancing techniques - the state-of-the-art', *International Journal Of Production Research*, vol. 30, no. 9, 2111-2135.

Appendix A

Feature:Process Matrix

	milling										drilling					
	cavity milling	chamfer milling	contouring	edge milling	face milling	shoulder milling	slot milling	surface milling	t slot milling	v slot milling	die threading	drilling	reaming	tapping	trepanning	countersinking
icy													1			
igv																
isp																
itd														1		
itp																
ipf																
esp																
egv																
ecy																
efa																
erg																
etp																
etd											1					
epf																
erg																
sf2			1					1								
sf3								1								
pfa			1	1	1			1								
psd			1			1										
pcf		1	1					1								
bho													1			
pcs					1											1
pcb																
pgv																
ppk	1															
pho													1		1	
cst							1									
pst							1									
pky									1							
vst										1						

	boring				turning											grinding				
																flat		centre		
	profile boring	taper boring	thread boring	lathe drilling	facing	cylindrical turning	grooving	chamfering	copy turning	die threading	parting	profile turning	taper turning	tapping	thread cutting	traverse grinding	plunge grinding	cylindrical grinding	face grinding	thread grinding
icy	1			1														1		
igv																		1		
isp	1																	1	1	
itd			1											1						1
itp	1	1																		
ipf	1																			
esp					1	1			1									1	1	
egv							1		1									1		
ecy						1			1									1		
efa					1				1		1					1			1	
erg							1		1										1	
etp								1	1				1							
etd									1	1					1					1
epf									1			1								
erg																				
sf2																				
sf3																				
pfa																1				
psd																1				
pcf																1				
bho																				
pcs																				
pcb																				
pgv																				
ppk																				
pho																				
cst																1				
pst																1				
pky																				
vst																				

Appendix B
Process quality limit data

Table 1: Economically attainable accuracy and surface finish for processes for external cylindrical surfaces

Process		Tolerance grade IT	Surface Roughness R_a (μm)
Turning	Rough turning	12-13	10-80
	Semifinish turning	10-11	2.5-10
	Finish turning	7-9	1.25-2.5
	Diamond turning	5-6	0.08-1.25
Groove turning	In one pass	11-12	10-20
	In two passes	10-11	2.5-10
Grinding	Rough grinding	7-9	0.63-2.5
	Semifinish grinding	6-7	0.16-0.63
	Finish grinding	5-6	0.08-0.16
Lapping	Semifinish lapping	5-6	0.04-0.64
	Finish lapping	3-5	0.008-0.08
Superfinishing		3-5	0.008-0.16
Polishing		3-5	0.008-1.25

Source: Wang, H-P, and Li, J-k: "Computer-Aided Process Planning, Advances in Industrial Engineering, 13", Elsevier, Amsterdam, Netherlands, 1991, p.106.

Table 2: Economically attainable accuracy and surface finish for processes for internal cylindrical surfaces

Process		Tolerance grade IT	Surface Roughness R_a (μm)
Drilling		11-13	5-80
Counterboring		10-11	1.25-20
Reaming	Rough reaming	8-9	1.25-5
	Semifinish reaming	7-8	0.63-1.25
	Finish reaming	6-7	0.16-0.63
Boring	Rough boring	12-13	5-20
	Semifinish boring	10-11	2.5-10
	Finish boring	7-9	0.63-1.25
	Diamond boring	5-7	0.16-0.63
Broaching	Semifinish broaching	9-10	0.32-2.5
	Finish broaching	6-9	0.16-0.63
Grinding	Rough grinding	7-9	0.63-1.25
	Semifinish grinding	6-7	0.16-0.63
	Finish grinding	5-6	0.08-0.16
Honing	Semifinish honing	6-7	0.16-1.25
	Finish honing	4-6	0.04-0.32
Lapping	Semifinish lapping	5-6	0.04-0.63
	Finish lapping	3-5	0.008-0.08
Superfinishing		3-5	0.008-0.16

Source: Wang, H-P, and Li, J-k: "Computer-Aided Process Planning, Advances in Industrial Engineering, 13", Elsevier, Amsterdam, Netherlands, 1991, p.106.

Table 3: Economically attainable accuracy and surface finish for processes for plane surfaces

Process		Tolerance grade IT	Surface Roughness R_a (μm)
Milling	Rough milling	11-13	5-20
	Semifinish milling	8-11	1.25-10
	Finish milling	6-8	0.32-1.25
Turning	Rough turning	12-13	10-80
	Semifinish turning	10-11	2.5-10
	Finish turning	7-9	1.25-2.5
	Diamond turning	6	0.08-1.25
Planing	Rough planing	11-13	5-20
	Semifinish planing	8-11	2.5-10
	Finish planing	6-8	0.63-5
Broaching	Semifinish broaching	10-11	0.63-2.5
	Finish broaching	6-9	0.16-0.63
Grinding	Rough grinding	7-9	0.63-1.25
	Semifinish grinding	6-7	0.16-0.63
	Finish grinding	5-6	0.08-0.16
Lapping	Semifinish lapping	5-6	0.04-0.63
	Finish lapping	3-5	0.008-0.08
Polishing		3-5	0.008-1.25
Superfinishing		3-5	0.008-0.16

Source: Wang, H-P, and Li, J-k: “Computer-Aided Process Planning, Advances in Industrial Engineering, 13”, Elsevier, Amsterdam, Netherlands, 1991, p.106.

Table 4: Machining Processes vs. Tolerance Grades

Machining Process	Tolerance grade (IT)									
	4	5	6	7	8	9	10	11	12	13
Lapping and Honing	■	■								
Cylindrical Grinding		■	■	■						
Surface Grinding		■	■	■	■					
Diamond Turning		■	■	■						
Diamond Boring		■	■	■						
Broaching		■	■	■	■					
Reaming			■	■	■	■	■			
Turning				■	■	■	■	■	■	■
Boring					■	■	■	■	■	■
Milling							■	■	■	■
Drilling							■	■	■	■
Planing and Shaping							■	■	■	■

Source: E. Oberg (*ed.*) "Machinery's Handbook", 24th edition, Industrial Press Inc., New York,1992, p. 607.

Table 5: Surface Roughness Average (micrometers) vs. Process

	Surface roughness average, R_a (μm)												
Process	50	25	12.5	6.3	3.2	1.6	0.8	0.4	0.2	0.1	.05	.025	.012
Sand Casting	▒	■	■	▒									
Die Casting					▒	■	■	▒					
Investment Casting				▒	■	■	▒	▒					
Perm. Mould Casting				▒	■	■	▒						
Cold Rolling				▒	■	■	■	■	▒	▒			
Hot Roing	▒	■	■	▒									
Forging		▒	■	■	■	▒							
Extruding			▒	▒	■	■	■	▒					
Flame Cutting	▒	■	■										
Sawing	▒	■	■	■	■	■	▒						
Snagging	▒	■	■	■	▒								
Drilling			▒	■	■	■	▒						
Reaming				▒	■	■	■	▒					
Planing, Shaping		▒	■	■	■	■	▒	▒					
Broaching					■	■	■	▒					
Milling		▒	▒	■	■	■	■	▒	▒				
Boring, Turning		▒	■	■	■	■	■	▒	▒	▒	▒		
Laser Cutting				▒	■	■	■	▒	▒				
Honing						▒	■	■	■	■	▒	▒	
Lapping							▒	■	■	■	■	▒	▒
Polishing							▒	■	■	■	▒	▒	▒

Key:

■	Process suitable for average application
▒	Process suitable for less frequent application

Source: E.Oberg (*ed.*), "Machinery's Handbook", 24th edition, Industrial Press Inc., New York, 1992, p. 672.

Appendix C

Process model details

This appendix lists the details of the individual process models in CESS. Each process model which has been developed is listed with a diagram, a short description and the equations used to calculate processing time. Each main process is divided up into several sub-classifications which allow the system to model the small differences in the process when it is used to create particular geometries, or is carried out on particular machine tools. For each of these sub-processes, there will be a further division into different quality levels.

1.1 Turning

The classification of turning processes includes both external and face turning, but not internal turning, which is classed as boring. These processes are characterised by the rotation of the workpiece, whilst the tool remains relatively static. The high cutting speeds required to deform the metal are achieved by the rotation of the workpiece, whilst the cutting tools is simply moved through the workpiece to describe the required surface contour. The advantages of this approach are two-fold: firstly, this is the most simple method of creating accurate axi-symmetric features, and secondly, more power can be used and thus higher cutting speeds.

The different classes of turning are illustrated in Figure 1.

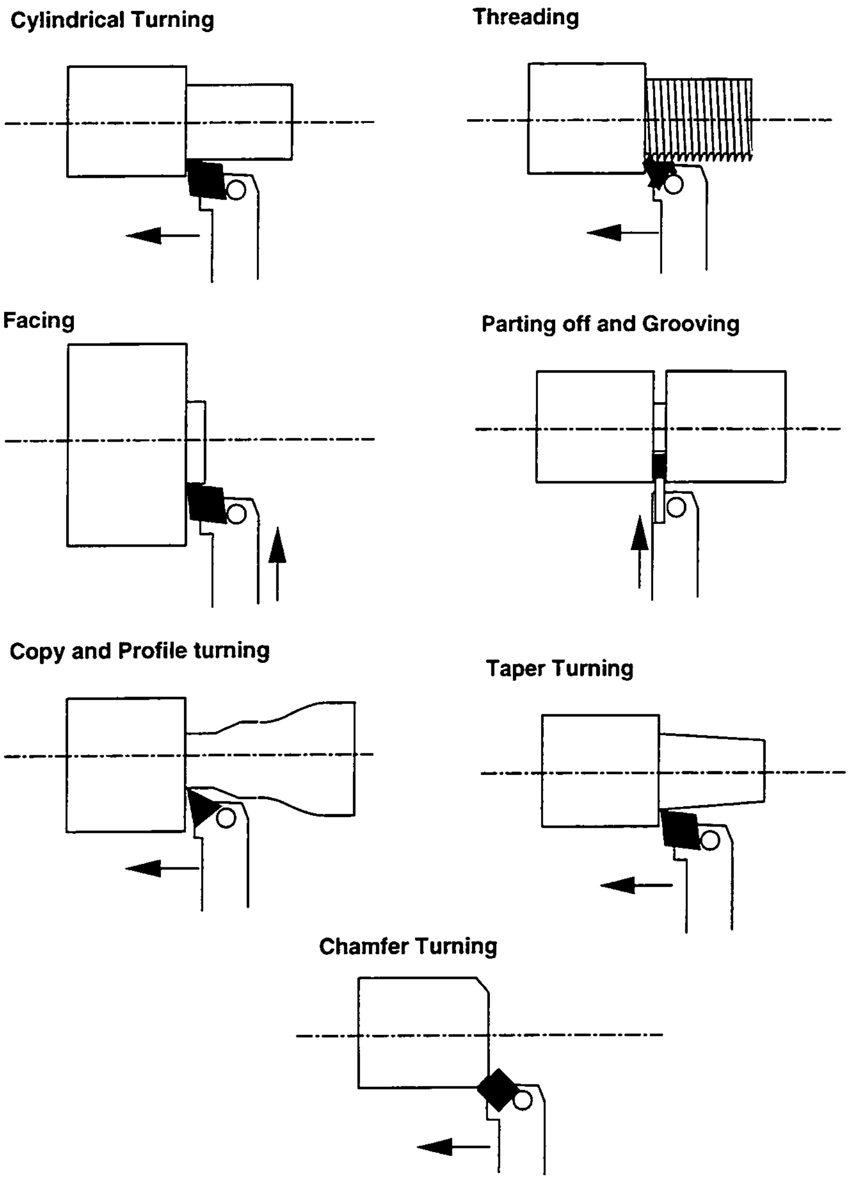


Figure 1: Turning sub-processes

1.1.1 Cylindrical turning.

In cylindrical turning, the tool is fed only in the axial direction, so that only cylindrical surfaces are generated. Processing time for cylindrical turning is calculated using the generic turning model to determine parameters, with time determined as:

$$t = p\frac{l.\pi.D}{1000.v.s}$$

where p = number of passes,

l = feature length (mm),

D = feature diameter (mm),

v = cutting velocity (m/min),

s = table feed (mm/rev),

t = time (min).

Table 1: Process limits for turning cylindrical surfaces

	Rough Turning	Semi Finish Turning	Semi Finish Turning	Diamond turning
Roughing process	n/a	Rough Turning	Semi Finish Turning	Semi Finish Turning
IT.min	12	9	7	5
IT.max	13	11	9	6
Ra.min	80.0	10.0	2.5	1.25
Ra.max	10.0	2.5	1.25	0.08
Ra.req	80.0	45.0	6.0	6.0
Ra.rough	80.0	6.0	2.5	n/a
Max. DOC	500.0	4.0	1.0	0.5
Ideal DOC	n/a	5.0	1.5	1.0

Table 2: Process limits for turning plane surfaces

	Rough Turning	Semi Finish Turning	Finish Turning	Diamond Turning
Roughing process	n/a	Rough turning	Semi finish turning	Finish turning
IT.min	12	10	6	6
IT.max	13	11	8	6
Ra.min	80.0	10.0	1.25	1.25
Ra.max	10.0	2.5	0.32	0.08
Ra.req	80.0	45.0	6.0	6.0
Ra.rough	45.0	6.0	n/a	n/a
Max. DOC	500.0	5.0	1.5	1.0
Ideal DOC	n/a	4.0	1.0	0.5

1.1.2 Facing

In face turning, or facing, the tool is fed only in one direction, perpendicular to the axis of rotation. This allows the generation of flat end surfaces. (N.B. The tool need only be fed across one half of the face to generate it since the workpiece is spinning).

$$t = p\frac{\pi.(d_o^2 - d_i^2)}{4000.v.s}$$

where d_o = outer diameter (mm)

d_i = inner diameter (mm),

1.1.3 Grooving

In grooving, a specialist insert is used to create narrow cylindrical grooves. This process is used when the groove is too narrow for a conventional cutting tool.

$$t = p\frac{\pi.(d_o^2 - d_i^2)}{4000.v.s}$$

1.1.4 Chamfering

The chamfering process is used to generate small tapers on the corners of cylindrical surfaces (chamfers). The angle is fixed (either 45 or 60 degrees), and the length is usually small, making the process distinct from taper turning, which is used to generate a wide range of angles over much greater lengths.

$$t = p\frac{l.\pi.D}{1000.v.s}$$

1.1.5 Profile turning

In profile turning, the axial and radial feeds are controlled together to move the cutter along any profile. The feed rates in either direction may be varied to create curves and tapers. The tool geometry must be selected to avoid interference with the workpiece and therefore triangular and kite shaped inserts are used.

$$t = p\frac{l.\pi.D}{1000.v.s}$$

1.1.6 Copy turning

Copy turning is the mechanical equivalent to the CNC process of profile turning. In this case, however, the, tool path is controlled by the movement of a guide sensor over a pattern shape. The pattern is typically a master copy of the desired component.

$$t = p\frac{l.\pi.D}{1000.v.s}$$

1.1.7 Parting

Parting follows the same format as grooving only the depth of cut is equal to the initial radius, so that the workpiece is cut into two pieces. The process model is therefore similar, although clearly only a single pass is necessary.

$$t = \frac{\pi.D^2}{4000.v.s}$$

1.1.8 Taper turning

In taper turning the tool is fed at a constant rate in both directions in order to generate a taper. Any angle may be generated by varying the relative rates of feed.

$$t = p\frac{l.\pi.D}{1000.v.s}$$

1.1.9 Thread cutting

The cutting of threads on a lathe uses a specialist cutting tool or insert. The rate of feed and speeds must be correctly set to generated the required pitch, and therefore parameter selection uses a different approach to the other turning processes.

$$t = \frac{l.\pi.D}{1000.v.p_i}$$

where p_i = thread pitch (mm).

1.2 Boring

Boring processes are basically the same as turning, only performed on the internal surface of cylindrical components. In order to perform a boring operation, an access hole must be present to allow entry of the tool. Therefore, boring is only suitable for enlarging existing holes. Boring is required, however, as it is capable of producing high quality dimensions of any size, whereas drilling and reaming can only make holes of sizes equal to the available tool sizes. In addition, boring can be used on large components where drill bits of an appropriate size are not available, for example where the internal diameter has been produced by a casting process. The boring classes are illustrated in Figure 2.

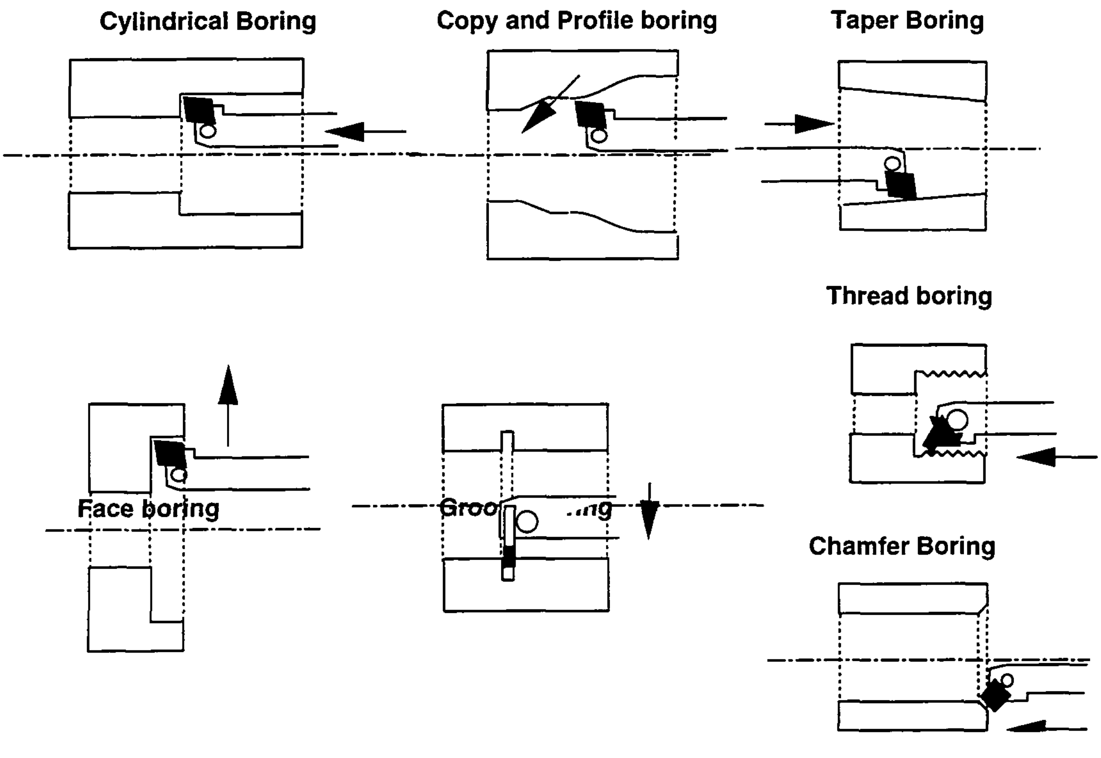


Figure 2: Boring sub-processes

Table 3: Process limits for boring

	rough boring	semi finish boring	finish boring	diamond boring
Roughing process	n/a	Rough Boring	Semi Finish Boring	Semi Finish Boring
IT.min	12	10	7	5
IT.max	13	11	9	7
Ra.min	20.0	10.0	2.5	0.63
Ra.max	5.0	2.5	0.63	0.16
Ra.req	80.0	13.0	6.0	6.0
Ra.rough	13.0	6.0	n/a	n/a
Max. DOC	500.0	5.0	1.5	1.0
Ideal DOC	n/a	4.0	1.0	0.5

1.2.1 Cylindrical boring

The internal equivalent of cylindrical turning, this process has the same time calculation method. It shares the additional constraints common to all boring operations, however, of requiring tool access. Therefore there is a minimum diameter hole which must be present initially.

$$t = p\frac{l.\pi.D}{1000.v.s}$$

1.2.2 Face boring

Face boring is the internal equivalent of facing. This process is used when the faces of internal features must be finished. The process time algorithm is the same as the external facing algorithm, except that the cutting distances are calculated from inside towards the outside.

$$t = p\frac{\pi.(d_o^2 - d_i^2)}{4000.v.s}$$

1.2.3 Groove boring

Groove boring is a limited process, as access is difficult for deep grooves, especially where the bore is a narrow diameter.

$$t = p\frac{\pi.(d_o^2 - d_i^2)}{4000.v.s}$$

1.2.4 Chamfer boring

Chamfer boring is used to create chamfers on internal edges of cylindrical parts.

$$t = p\frac{l.\pi.D}{1000.v.s}$$

1.2.5 Profile boring

Profile boring is the same process as profile turning, except that the cutting tool is a boring bar instead of the simple turning tool. This allows internal surfaces to be generated.

$$t = p\frac{l.\pi.D}{1000.v.s}$$

1.2.6 Copy boring

Copy boring is the equivalent of copy turning, except that it generates only internal surfaces, using a boring bar.

$$t = p\frac{l.\pi.D}{1000.v.s}$$

1.2.7 Taper boring

Taper boring is used to generate internal tapers, in exactly the same way as taper turning.

$$t = p\frac{l.\pi.D}{1000.v.s}$$

1.2.8 Thread boring

Thread boring is used to cut internal threads, in the same way that thread cutting is used for external threads. The time algorithm is therefore the same.

$$t = \frac{l.\pi.D}{1000.v.p_i}$$

1.3 Drilling

Drilling (Figure 3) may be thought of as a sub-set of milling, since the process bears many similarities. It is, however, a sufficiently different process to merit a separate classification. The drilling process consists of moving an axi-symmetric cutting tool along its axis of symmetry into the workpiece, to form a cylindrical hole. The cutting tool has a number of cutting edges on the bottom and sides, and a means of removing the workpiece material as chips. In general drilling machines operate vertically, although in CNC machining centres or manually, drilling may be performed at any orientation. There are several sub-processes within the class of drilling, each of which produces a slightly different feature set using a modified tool. All drilling sub-processes use the same generic drilling time algorithm, with the differences between the types being implemented through the use of alternative cutting conditions, and through the method which calculates the depth of cut of the operation.

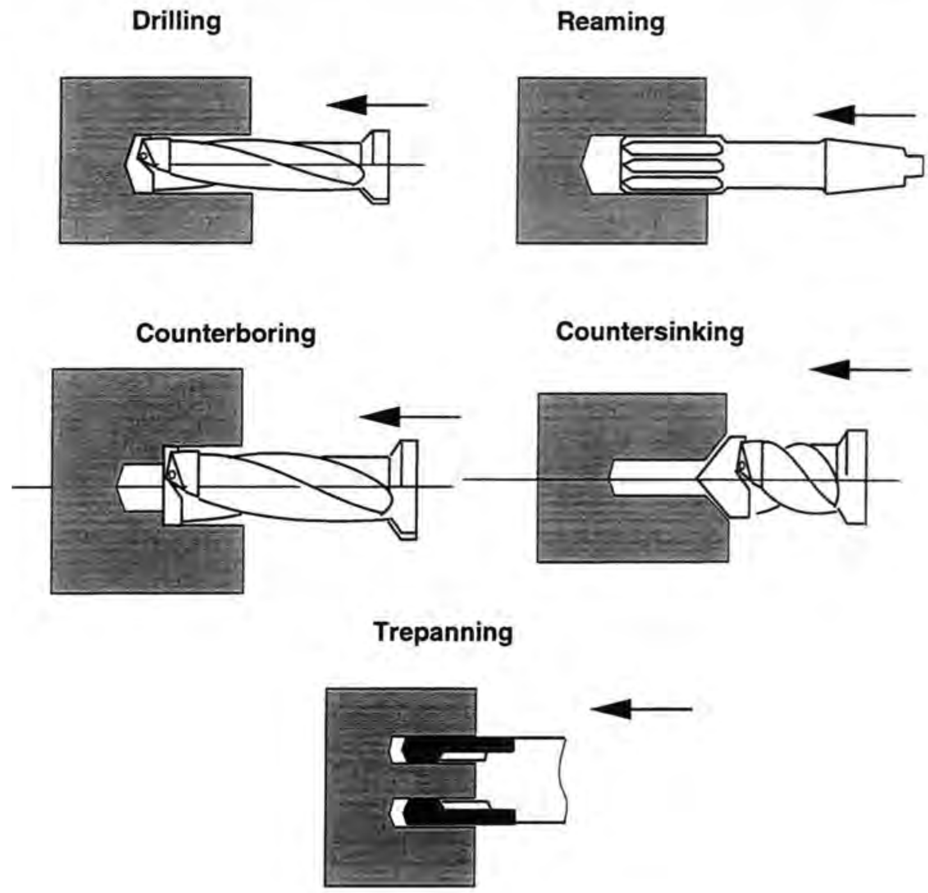


Figure 3: Drilling sub-processes

The important variable for the drilling sub-processes is the effective depth of cut, a_p, which determines the power requirement for a given operation. For solid drilling, a_p is equal to the radius of the hole. For other processes, a_p is the depth of material removed.

Table 4: Process limits for drilling sub-processes

	Drilling	Counterboring	Counter-sinking	Reaming	Trepanning
Roughing process	n/a	n/a	Drilling	Drilling	n/a
IT.min	11	10	10	6	10
IT.max	13	11	11	9	11
Ra.min	80.0	20.0	20.0	5.0	20.0
Ra.max	5.0	1.25	1.25	0.16	1.25
Ra.req	80.0	43.0	43.0	10.0	43.0
Ra.rough	43.0	20.0	20.0	n/a	20.0
Max DOC	80	n/a	n/a	1.0	120
Ideal DOC	n/a	n/a	n/a	1.0	n/a

1.3.1 Solid Drilling

In solid drilling, there is no existing hole in which to drill into. Therefore the depth of cut parameter, a_p, is equal to the radius of the hole feature. Drilling cutting parameters are provided by the materials database.

1.3.2 Counterboring

Counterboring is the term for drilling where there is already a hole present. The aim is to enlarge the hole or to form a stepped hole. As in the case of reaming, the depth of cut value a_p is the difference between the radii of the starting and finishing holes. The aggregate process model for counterboring uses the same cutting conditions as for drilling. This is acceptable whilst the difference in diameters between the holes is large enough for a reasonable depth of cut. If the holes are too similar in size, counterboring is unsuitable for reasons of positional accuracy.

1.3.3 Countersinking

Countersinking is the creation of chamfered holes, where the diameter changes gradually from one size to another. This is a process which can use very aggressive data since the tools position is held in place. The effective a_p for counterboring is taken as the half the true depth of cut, to reflect the shape of the tool.

1.3.4 Reaming

Reaming is a finishing process for holes, which requires that the feature is very near its finished shaped before use. That is, the maximum depth of cut is very low. For reaming, the depth of cut a_p equates to the thickness of the material to be removed. Since the power will always be low in reaming, the standard material cutting conditions (s and V_c) will tend to apply. In reaming these conditions are less severe than drilling.

1.3.5 Trepanning

In the trepanning process, the tool has a hollow core, which causes it to cut a ring-shaped hole. When used to cut thin material, trepanning is a more efficient alternative to solid drilling for creating large holes. When used for blind holes, circular grooves can be generated.

1.3.6 Tapping

Tapping is used to generate internal threads by cold forming the surface metal of the hole. Cutting speeds for tapping are much slower than for drilling. Downfeed is equal to the pitch of the thread.

1.4 Milling

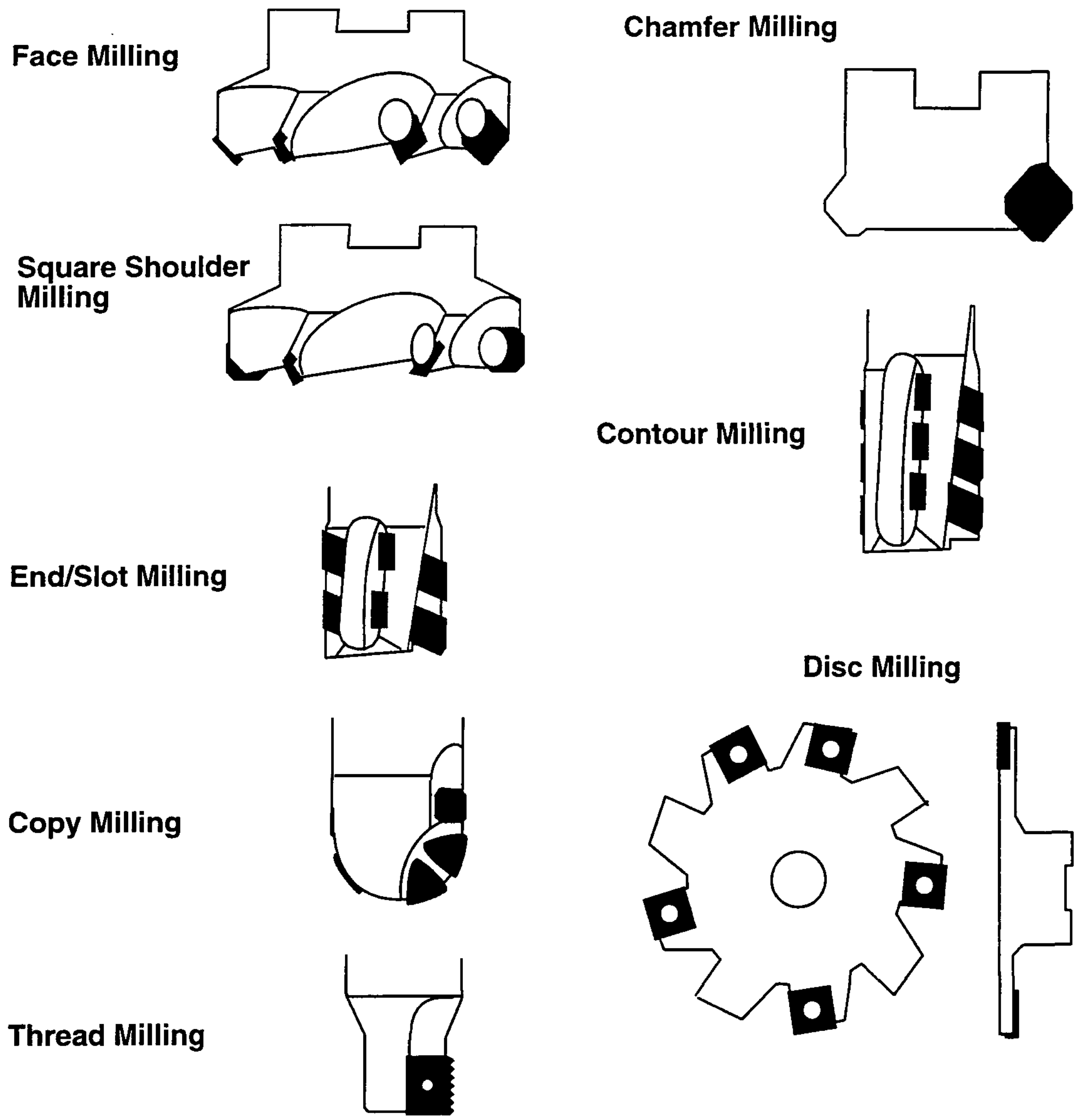


Figure 4: Cutting tools for milling sub-processes

The milling class is divided into a number of sub-classes depending on the particular type of cutting tool being used, since this will determine the set of features which that process can make (Figure 4). The process selection matrix for the features has been set up so that the most appropriate of the various milling sub-classes is considered, rather than suggesting less than ideal milling tools in addition to the more suitable type. For example, when facing a component, a slotting milling tool could be used, but would always prove less suitable than the facing tool. Therefore, the system is designed not to suggested the slotting tool in this case. The system uses the same process quality limits for each of the milling process.

Table 5: Process limits for milling quality levels

	Rough Milling	Semi Finish Milling	Finish Milling
Roughing process	n/a	Rough Milling	Semi Finish Milling
IT.min	11	8	6
IT.max	13	11	8
Ra.min	20.0	10.0	1.25
Ra.max	5.0	1.25	0.32
Ra.req	80.0	13.0	6.0
Ra.rough	13.0	6.0	n/a
Max. DOC	5000.0	4	2
Ideal DOC	n/a	4	1

Table 6: Tool limits according to process type

	a_{amax}	a_{rmax}
Shoulder milling	88	32
Slot milling	65	50
Surface milling	32	32
Bore milling	20	10
Cavity milling	40	32
Chamfer milling	88	32
Disc milling	150	30.5
Edge milling	88	32
Face milling	19	200

1.4.1 Face milling

The face milling model is the simplest milling model. The typical face milling tool has a large diameter and number of inserts, but there is only one level of inserts. This means that the maximum radial depth of cut is high, but the axial depth of cut is low. It is important to note that the model follows the standard recommended use of the tools in only allow a tool diameter of 40%, to give an efficient engagement angle.

1.4.2 Slot milling

In slot milling, the tool width is constrained by the size of the feature. Since it will usually be the case that finishing passes are required, and it is unlikely that there will be a tool of exactly the required width, it is possible to assume that there will always be at least two radial passes, and this is forced in the algorithm by the use of half the feature width for the maximum axial radius.

1.4.3 Cavity milling

In cavity milling, an end milling cutting is typically used. Cavity milling is a process which has been devised to model the cutting of pockets: typically such an activity requires multiple cutting passes and potentially more than one size of cutting tool. If the pocket is closed, then the tool must be plunged into the workpiece in some way; plunging is either vertical, using a drill/mill, or by ramping the tool in at an angle. The latter approach allows a wider tool selection and is more gentle. Ramping can be performed whilst circling the tool for small pockets, or linearly, for larger pockets. In all these cases, the depth of cut changes significantly during a pass, and therefore a generic model is difficult to devise.

1.4.4 Chamfer milling

The chamfer milling process uses a similar tool to a shoulder mill. The cutting teeth are set at the angle of the chamfer, however, allowing a simple generation of the chamfer without complex control of the cutting head, which would be required to create the chamfer using an end miller. The time algorithm for chamfer milling takes account of the reduced cross-section of material being removed in a pass.

1.4.5 Thread milling

In thread milling, a special tool which reflects the thread geometry is used. The thread is generated by helical interpolation around the diameter.

1.4.6 Copy milling

The copy milling process, in contrast to copy turning, is a CNC process. Copy milling uses a small diameter, ball nosed cutter which can be used to generate a wide variety of surfaces (all except square shoulders). This allows the use of a single tool for complex shapes.

1.4.7 Contour milling

In contour milling, a helical edge milling tool is used to generate a surface which is curved in one direction. The tool path must be numerically controlled to move two axis independently.

1.4.8 Shoulder milling

In shoulder milling, the tool has open access to the feature, and therefore large tool sizes can be selected.

1.4.9 Disc milling

In disc milling, the cutting tool is disc shaped, with the cutting teeth on the edge. This process is used for creating narrow through slots and for parting components.

1.4.10 Double angle milling

Similar to disc milling, double angle milling uses a disc shaped cutter, with the difference being that the teeth are shaped to leave a v-shaped notch in the material instead of a squared notch. The time calculation and constraint methods are essentially the same as for disc milling.

1.4.11 T-Slot milling

The milling of T-slots requires a two stage procedure, or a specialist tool. The former approach is the more flexible since a variety of slot sizes may be produced. In this case, the first pass produce the "neck" slot, allowing room for access with a wider cutter to generate the main part of the slot.

1.4.12 Bore milling

In bore milling, cylindrical surfaces are generated by moving an edge-milling cutter in circles. This method can be used to generate holes much larger than the size of the milling cutter, and is a useful alternative to boring for the creation of large diameter holes, although maximum quality is not quite as high. The process limits for bore milling are different than the other milling processes, since cylindrical surfaces are being generated.

Table 7: Process limits for bore milling

	Rough Bore milling	Semi Finish Bore Milling	Finish Bore Milling
Roughing process	n/a	Rough Milling	Semi Finish Milling
IT.min	11	8	6
IT.max	13	11	8
Ra.min	20.0	10.0	1.25
Ra.max	5.0	1.25	0.32
Ra.req	80.0	13.0	6.0
Ra.rough	13.0	6.0	n/a
Max. DOC	200.0	10	3
Ideal DOC	n/a	6	2.5

1.5 Grinding

Grinding processes consist of moving an abrasive surface against the workpiece to remove the material in a gradual process. Grinding is not suitable for removal of a large amount of material since the cost per kg of material is very high. However, high quality surfaces finishes can be achieved relatively easily using a grinding process. There are a number of sub-classes of grinding which relate to either the feature type which can be made, or the mechanics of the process itself which affect the quality or the processing time. Each process can be detailed separately within the system.

Table 8: Process limits for surface grinding

	Rough Grinding	Semi Finish Grinding	Finish Grinding
Roughing process	Semi Finish Milling	Rough Grinding	Semi Finish Grinding
IT.min	7	6	5
IT.max	9	7	6
Ra.min	2.5	0.63	0.16
Ra.max	0.63	0.16	0.08
Ra.req	50.0	1.5	0.40
Ra.rough	1.5	0.4	0.12
Max. DOC	0.5	0.15	0.1
Ideal. DOC	0.2	0.14	0.07
Max DOC / pass	0.07	0.032	0.013

Surface grinding can be performed either as plunge or traverse grinding. In plunge grinding the grinding wheel is fed down onto the workpiece only, whereas in traverse grinding it moves across the workpiece at the same time. CESS models the traverse grinding process.

$$t = p\frac{w.l}{ww'.v.1000}$$

1.5.1 Cylindrical grinding

Cylindrical grinding, like surface grinding, can be performed in either the plunge or traverse mode. The CESS model assumes that traverse grinding will be used. Internal and external surfaces may be produced. The equation for time is the same in each case and was shown in Chapter 5.

Table 9: Process limits for external cylindrical grinding

	Rough	Semi Finish	Finish
Roughing process	Semi Finish Turning	Rough cyl. Grinding	Semi Finish cyl. Grinding
IT.min	7	6	5
IT.max	9	7	6
Ra.min	2.5	0.63	0.16
Ra.max	0.63	0.16	0.08
Ra.req	50.0	1.5	0.40
Ra.rough	1.5	0.4	0.12
Max. DOC	0.5	0.15	0.1
Ideal. DOC	0.2	0.14	0.07
Max DOC / pass	0.07	0.032	0.013

Table 10: Process limits for internal cylindrical grinding

	rough int grinding	semi finish int grinding	finish int grinding
Roughing process	Drilling	Rough Int Grinding	Rough Int Grinding
IT.min	7	6	5
IT.max	9	7	6
Ra.min	2.5	0.63	0.16
Ra.max	0.63	0.16	0.08
Ra.req	50.0	1.5	0.4
Ra.rough	1.5	0.4	0.12
Max. DOC	0.5	0.15	0.1
Ideal. DOC	0.2	0.14	0.07
Max DOC / pass	0.07	0.032	0.013

1.6 Precision abrasion

The precision abrasion class covers a set of abrasive processes which are designed to produce very high surface finishes. These processes are lapping, honing, polishing and superfinishing. Each has slightly different characteristics, and can achieve a different level of surface finish. Precision processes may be necessary where normal grinding cannot achieve the desired result economically. Although CESS does not currently implement process time calculations for these process, the capabilities of the processes have been modelled.

Table 11: Precision abrasion of plane surfaces

	Lapping	Superfinishing	Polishing
Roughing process	Semi Finish Grinding	Finish Grinding	Finish Grinding
IT.min	3	3	4
IT.max	6	5	5
Ra.min	0.63	0.16	1.25
Ra.max	0.008	0.008	0.008
Ra.req	0.40	0.12	0.12
Ra.rough	n/a	n/a	n/a

Table 12: Precision abrasion of internal cylindrical surfaces

	Lapping	Superfinishing	Honing
Roughing process	Finish Int Grinding	Finish Int Grinding	Finish Int Grinding
IT.min	3	3	4
IT.max	6	5	7
Ra.min	0.63	0.16	1.25
Ra.max	0.008	0.008	0.04
Ra.req	0.12	0.12	0.4
Ra.rough	n/a	n/a	n/a

1.6.1 Honing

This is a finishing process which uses abrasive stones to produce a very smooth finish on cylindrical surfaces.

1.6.2 Polishing

This is a finishing process which can be used for prismatic or axi-symmetric features. A soft wheel is used to apply a fine abrasive in particulate form.

1.6.3 Superfinishing

Super-finishing is a proprietary finishing process which can be used for prismatic or axi-symmetric features.

1.6.4 Lapping

Lapping uses a soft metal tool with abrasive particles to produce very high surface finishes. Lapping can be adapted for either cylindrical or plane surfaces.

1.7 Broaching

In broaching, multi-point cutting tool is driven in a straight line through the workpiece in order to generate a hole or slot of the same profile as the cutting tool. Broaching can produce complex shapes by using specialist tools, but it does not give a high quality finish, and is relatively uneconomic, except for small features. Since broaching requires specialist tools, it has not been included in the CESS system at this time.

Table 13: Broaching quality limits

Roughing process	Counterboring
IT.min	6
IT.max	11
Ra.min	2.5
Ra.max	0.16
Ra.req	11.0
Ra.rough	n/a

1.8 Sawing

In sawing, a multi-toothed band is reciprocated across the workpiece to split it into two parts. This process is used mainly in preparation of billets for manufacturing. Since the CESS routing algorithm assumes that the raw material is available in billet form, it was judged that it was not necessary to develop an aggregate process model for sawing.

1.9 Planing

In planing, a single cutting tool is repeatedly driven through the workpiece in a straight line, removing a thin layer of material each pass. Whilst planing is the linear equivalent of turning, the fact that it is discontinuous (requiring multiple entries into the material) means that the characteristics of the process are very different, and it is not suited to high volume work or the production of high tolerances. Planing is rarely used since generally milling is a more efficient and capable process. It was therefore decided that it would be unnecessary to model planing for CESS.

Table 14: Planing quality limits

	Rough Planing	Semi Finish Planing	Finish Planing
Roughing process		Rough Planing	Semi Finish Planing
IT.min	11	8	6
IT.max	13	11	8
Ra.min	20.0	10.0	5.0
Ra.max	5.0	2.5	0.63
Ra.req	50.0	13	6.0
Ra.rough	13.0	6.0	n/a

Appendix D
Material database with sample cutting conditions

		General		Milling	Turning			Drilling		
Name	Desc	Density	cost	*Vp*	*v*	*f*	k_{sm}	dv_c	*df*	*tvc*
mildsteel	mildsteel	7.83	1.45	25.0	355.0	0.20	1900	100	0.22	167
steel 080M15	unalloyed steel (C=.15%)	7.83	1.45	25.0	355.0	0.20	1900	100	0.22	167
steel 080M46	unalloyed steel (C=.35%)	7.83	1.50	23.0	305.0	0.20	2100	92	0.23	167
steel 070M55	unalloyed steel (C=.60%)	7.83	1.55	21.0	290.0	0.20	2250	92	0.23	167
steel 805M20	non-hardened low alloy steel	7.83	1.60	21.0	215.0	0.20	2100	90	0.22	133
steel 816M40	hard ened low alloysteel	7.83	1.65	17.0	120.0	0.20	2700	66	0.22	133
steel 832M13	annealed high alloy steel	7.83	1.70	19.0	190.0	0.20	2600	68	0.20	133
hard 832M13	hardened high alloy steel	7.83	1.75	17.0	100.0	0.20	3900	58	0.20	133
cast steel	unalloyed cast steel	7.83	1.60	27.0	165.0	0.20	2000	92	0.23	120
cast alloy	low alloyed cast steel	7.83	1.65	24.0	120.0	0.20	2500	75	0.21	120

Vc Specific material removal rate for milling (cm^3/min.kW),

v Preferred cutting speed for turning (m/min),

f Preferred feed rate for turning (mm/rev),

K_{sm} Specific resistance to cut for turning (N/mm^2),

dv_c Preferred cutting speed for drilling (m/min),

df Preferred down feed rate for drilling (mm/rev),

tv_c Preferred cutting speed for threading (m/min).

Appendix E

Product models used in testing

Table 1: Full moving armature product model

Name	PARENTS	PROPLIST	VALUES
order827426050	order827426050	customer product	Hugh ERD30
ERD30	products order827426050	selfname typeclass	ERD30 products
Moving_Armature	components ERD30	amount selfname typeclass material	1 Moving_Armature components mild steel
sf2872166221	sf2 Moving_Armature	component number rank selfname set-up typeclass prefix it roughness	Moving_Armature 3 Unknown sf2872166221 1 features sf2 Unknown 1.5
depth_sf2872166221	depth sf2872166221	upper lower nominal typeclass	0.5 0.5 6.0 geometry
width_sf2872166221	width sf2872166221	upper lower nominal typeclass	0.3 0.3 10.0 geometry
length_sf2872166221	length sf2872166221	upper lower nominal typeclass	0.2 0.2 8.0 geometry
efa872166099	efa Moving_Armature	component number rank selfname set-up typeclass prefix it roughness	Moving_Armature 1 Unknown efa872166099 2 features efa Unknown 1.5

Name	PARENTS	PROPLIST	VALUES
diameter_efa872166099	diameter efa872166099	upper lower nominal typeclass	1.0 1.0 100.0 geometry
length_efa872166099	length efa872166099	upper lower nominal typeclass	1.0 1.0 1.0 geometry
cylinder827426233	cylinder Moving_Armature	diameter length parent selfname typeclass	101.0 6.0 Moving_Armature cylinder827426233 features
icy827756050	icy Moving_Armature	component number rank selfname set-up typeclass prefix it roughness	Moving_Armature 1 Unknown icy827756050 1 features icy Unknown Unknown
length_icy827756050	length icy827756050	upper lower nominal typeclass	0.5 0.0 6.0 geometry
diameter_icy827756050	diameter icy827756050	upper lower nominal typeclass	0.1 0.0 31.0 geometry
pho827756091	pho Moving_Armature	component number rank selfname set-up typeclass prefix it roughness	Moving_Armature 3 Unknown pho827756091 1 features pho Unknown Unknown
length_pho827756091	length pho827756091	name upper lower nominal typeclass	length 0.5 0.0 6.0 geometry
diameter_pho827756091	diameter pho827756091	upper lower nominal typeclass	0.15 0.0 8.45 geometry

Name	PARENTS	PROPLIST	VALUES
ecy827756181	ecy Moving_Armature	component number rank selfname set-up typeclass prefix it roughness	Moving_Armature 1 Unknown ecy827756181 1 features ecy Unknown Unknown
length_ecy827756181	length ecy827756181	upper lower nominal typeclass	0.1 0.0 6.0 geometry
diameter_ecy827756181	diameter ecy827756181	upper lower nominal typeclass	0.5 0.5 101.0 geometry
efa872162220	efa Moving_Armature	component number rank selfname set-up typeclass prefix it roughness	Moving_Armature 1 Unknown efa872162220 1 features efa Unknown Unknown
length_efa872162220	length efa872162220	upper lower nominal typeclass	0.1 0.1 1.0 geometry
diameter_efa872162220	diameter efa872162220	upper lower nominal typeclass	0.1 0.1 100.0 geometry

Table 2: Full flange component product model

Name	PARENTS	PROPLIST	VALUES
order852552184	order852552184	customer product	hugh prodone
prodone	products order852552184	amount selfname typeclass	Unknown prodone products
flange	components prodone	amount selfname typeclass material	1 flange components mild steel
itp860605150	itp flange	component number rank selfname set-up typeclass prefix it roughness	flange 1 Unknown itp860605150 1 features itp Unknown Unknown
angle_itp860605150	angle itp860605150	upper lower nominal typeclass	2.0 2.0 45.0 geometry
diameter_itp860605150	diameter itp860605150	upper lower nominal typeclass	0.5 0.5 299.0 geometry
length_itp860605150	length itp860605150	upper lower nominal typeclass	0.5 0.5 2.0 geometry
itp860605028	itp flange	component number selfname set-up typeclass prefix it roughness	flange 1 itp860605028 1 features itp Unknown Unknown
angle_itp860605028	angle itp860605028	upper lower nominal typeclass	2.0 2.0 45.0 geometry

Name	PARENTS	PROPLIST	VALUES
diameter_itp860605028	diameter itp860605028	upper lower nominal typeclass	0.5 0.5 299.0 geometry
length_itp860605028	length itp860605028	upper lower nominal typeclass	0.5 0.5 2.0 geometry
etp860604976	etp flange	component number selfname set-up typeclass prefix it roughness	flange 1 etp860604976 1 features etp Unknown Unknown
angle_etp860604976	angle etp860604976	upper lower nominal typeclass	2.0 2.0 45.0 geometry
diameter_etp860604976	diameter etp860604976	upper lower nominal typeclass	0.5 0.5 393.0 geometry
length_etp860604976	length etp860604976	upper lower nominal typeclass	0.5 0.5 7.0 geometry
etp860604879	etp flange	component number selfname set-up typeclass prefix it roughness	flange 1 etp860604879 1 features etp Unknown Unknown
angle_etp860604879	angle etp860604879	upper lower nominal typeclass	2.0 2.0 45.0 geometry
diameter_etp860604879	diameter etp860604879	upper lower nominal typeclass	0.5 0.5 393.0 geometry

Name	PARENTS	PROPLIST	VALUES
length_etp860604879	length etp860604879	upper lower nominal typeclass	0.5 0.5 7.0 geometry
efa859480299	efa flange	component number selfname set-up typeclass prefix it roughness	flange 1 efa859480299 1 features efa Unknown Unknown
diameter_efa859480299	diameter efa859480299	upper lower nominal typeclass	0.5 0.5 407.0 geometry
length_efa859480299	length efa859480299	upper lower nominal typeclass	0.5 0.5 5.0 geometry
efa859480256	efa flange	component number rank selfname set-up typeclass prefix it roughness	flange 1 Unknown efa859480256 1 features efa Unknown Unknown
diameter_efa859480256	diameter efa859480256	upper lower nominal typeclass	0.5 0.5 302.0 geometry
length_efa859480256	length efa859480256	upper lower nominal typeclass	0.5 0.5 5.0 geometry
icy859478025	icy flange	component number selfname set-up typeclass prefix it roughness	flange 1 icy859478025 1 features icy Unknown Unknown

Name	PARENTS	PROPLIST	VALUES
diameter_icy859478025	diameter icy859478025	upper lower nominal typeclass	0.5 0.5 295.0 geometry
length_icy859478025	length icy859478025	upper lower nominal typeclass	0.5 0.5 32.0 geometry
Cylinder859476088	cylinder flange	diameter length selfname typeclass	407.0 32.0 Cylinder859476088 features
ecy859476095	ecy flange	component number selfname set-up typeclass prefix it roughness	flange 1 ecy859476095 1 features ecy Unknown Unknown
length_ecy859476095	length ecy859476095	upper lower nominal typeclass	0.5 0.5 15.0 geometry
diameter_ecy859476095	diameter ecy859476095	upper lower nominal typeclass	0.5 0.5 407.0 geometry
esp859476243	esp flange	component number selfname set-up typeclass prefix it roughness	flange 1 esp859476243 1 features esp Unknown Unknown
length_esp859476243	length esp859476243	upper lower nominal typeclass	0.5 0.5 10.0 geometry
diameter_esp859476243	diameter esp859476243	upper lower nominal typeclass	0.5 0.5 393.0 geometry

Name	PARENTS	PROPLIST	VALUES
esp859476923	esp flange	component number selfname set-up typeclass prefix it roughness	flange 1 esp859476923 1 features esp Unknown Unknown
length_esp859476923	length esp859476923	upper lower nominal typeclass	0.25 0.25 3.0 geometry
diameter_esp859476923	diameter esp859476923	upper lower nominal typeclass	0.5 0.5 302.0 geometry
bho859478624	bho flange	component number selfname set-up typeclass prefix it roughness	flange 12 bho859478624 1 features bho Unknown Unknown
diameter_bho859478624	diameter bho859478624	upper lower nominal typeclass	0.1 0.1 25.0 geometry
length_bho859478624	length bho859478624	upper lower nominal typeclass	0.5 0.5 24.0 geometry
itd860602109	itd flange	component number rank selfname set-up typeclass prefix it roughness	flange 1 Unknown itd860602109 1 features itd Unknown Unknown
length_itd860602109	length itd860602109	upper lower nominal typeclass	0.5 0.5 32.0 geometry

Name	PARENTS	PROPLIST	VALUES
diameter_itd860602109	diameter itd860602109	upper lower nominal typeclass	0.5 0.5 295.0 geometry
pitch_itd860602109	pitch itd860602109	upper lower nominal typeclass	0.25 0.25 3.0 geometry

Table 3: Example prismatic component with two features

Name	PARENTS	PROPERTIES	VALUES
order875102747	order875102747	customer product	hugh ERD30
ERD30	products order875102747	amount selfname typeclass	Unknown ERD30 products
acube	components ERD30	amount selfname typeclass material	1 acube components mild steel
bho875574912	bho acube	component number selfname setup typeclass prefix it roughness	acube 1 bho875574912 1 features bho 12 40.0
length_bho875574912	length bho875574912	upper lower nominal typeclass	1.0 1.0 40.0 geometry
diameter_bho87557491 2	diameter bho875574912	upper lower nominal typeclass	0.05 0.05 8.0 geometry
pst875574810	pst acube	component number selfname setup typeclass prefix it roughness	acube 1 pst875574810 1 features pst 11 4.0
length_pst875574810	length pst875574810	upper lower nominal typeclass	0.1 0.1 200.0 geometry
depth_pst875574810	depth pst875574810	upper lower nominal typeclass	0.05 0.05 10.0 geometry

Name	PARENTS	PROPERTIES	VALUES
width_pst875574810	width pst875574810	upper lower nominal typeclass	0.5 0.5 50.0 geometry
prism875102774	prism acube	length width breadth selfname typeclass	200.0 200.0 200.0 prism875102774 features

Table 4: Example axi-symmetric part with three features

Name	PARENTS	PROPLIST	VALUES
order827426050	order827426050	customer product	Hugh ERD30
ERD30	products order827426050	selfname typeclass	ERD30 products
Moving_Armature	components ERD30	amount parent selfname typeclass material	1 ERD30 Moving_Armature components mildsteel
itd875577220	itd Moving_Armature	component number rank selfname setup typeclass prefix it roughness	Moving_Armature 1 Unknown itd875577220 1 features itd Unknown Unknown
pitch_itd875577220	pitch itd875577220	upper lower nominal typeclass	0.1 0.1 1.8 geometry
diameter_itd875577220	diameter itd875577220	upper lower nominal typeclass	1.0 1.0 20.0 geometry
length_itd875577220	length itd875577220	upper lower nominal typeclass	0.5 0.5 50.0 geometry
icy875576021	icy Moving_Armature	component number rank selfname setup typeclass prefix it roughness	Moving_Armature 1 Unknown icy875576021 1 features icy 12 2.0
diameter_icy875576021	diameter icy875576021	upper lower nominal typeclass	0.05 0.05 20.0 geometry

Name	PARENTS	PROPLIST	VALUES
length_icy875576021	length icy875576021	upper lower nominal typeclass	1.0 1.0 100.0 geometry
esp872162220	esp Moving_Armature	component number rank selfname setup typeclass prefix it roughness	Moving_Armature 1 Unknown esp872162220 2 features esp Unknown Unknown
diameter_esp872162220	diameter esp872162220	upper lower nominal typeclass	0.5 0.5 40.0 geometry
length_esp872162220	length esp872162220	upper lower nominal typeclass	0.5 0.5 50.0 geometry
cylinder827426233	cylinder Moving_Armature	diameter length parent selfname typeclass	50.0 100.0 Moving_Armature cylinder827426233 features

Appendix F

Factory models used in testing

Factory database

Factory	Cells	xext	yext
Warner_Electric	Drill_shop ERD ERS Jobbing Coils Friction Press_Shop	100.00	60.00
Factory B	MachineShop FabricationShop	50.00	40.00

Warner Electric Machines database

Cell	Machine tools	xext	yext	xcoord	ycoord	available	trate
Jobbing	DeanSmithGrace Herbert7b Pullomax Herbert4 Herbert2D Lapmaster Keithley Hobbing1 Hobbing2 Ajax	40	20	0	40	TRUE	5.5
Friction	Hardinge Herbert7a	40	20	0	40	TRUE	5.5
Press_Shop	Hi_Ton Branson	40	20	0	0	TRUE	4.5
Coils	Aumann0 Westminster Eubanks Marsilli	40	20	0	40	TRUE	5.5
Moulding	Daniels1 Daniels0	40	20	0	0	TRUE	4.5
Drill_shop	MultiDrill Tapping2 Tapping1 Tapping0	40	20	0	40	TRUE	5.5
ERS	Wadkin Warner_Swasey Traub0 Mazak1 Mazak0	40	20	0	40	TRUE	5.5
ERD	MazakAGV18a MazakAGV18b Mazak3 Mazak2 Traub1 Blanchard Amada	40	20	0	40	TRUE	5.5

Warner Electric Machines database (Part 1)

Name	model	type	cell	Tbatch	Tpiece
Keithley	Cylindrical grinding machine	centre_grinders	Jobbing	60.00	0.120
Hobbing1	Hobbing machine	hobbing_mc	Jobbing	30.00	0.200
Hobbing2	Hobbing machine	hobbing_mc	Jobbing	60.00	0.000
Ajax	Ajax Universal Mill	manual_milling_mc	Jobbing	20.00	2.000
Schenk	Balancing machine	centre_grinders	Jobbing	0.00	1.000
Hardinge	Chucking Lathe with extraction	manual_lathes	Friction	0.00	0.200
Herbert7a	Lathe with extraction	manual_lathes	Friction	0.00	0.200
Hi_Ton	Press	presses	Press_Shop	20.00	0.100
Branson	Ultrasonic Welder	welding_mc	Press_Shop	20.00	1.000
Marsilli	WM15 Twin spindle winding m/c	winding_mc	Coils	40.00	0.100
Eubanks	Wire Stripper	winding_mc	Coils	30.00	0.050
Westminster	Bonded Coil winding m/c	winding_mc	Coils	60.00	1.100
Tapping0	Tapping m/c	taps	Drills	20.00	0.200
Tapping1	Tapping m/c	taps	Drills	150.00	0.200
Tapping2	Tapping m/c	taps	Drills	150.00	0.200
MultiDrill	4 Head multi-station drill	column_drills	Drills	30.00	3.000
Lapmaster	Lapping m/c	lapping_mc	Jobbing	120.00	1.000
Daniels0	Transfer moulding m/c	moulding_mc	Moulding	40.00	0.100
Daniels1	Transfer moulding m/c	moulding_mc	Moulding	120.00	0.200
Pullomax	Nibbler	nibbling_mc	Jobbing	50.00	0.310
Herbert4	Lathe	manual_lathes	Jobbing	120.00	0.300
Herbert2D	Lathe	manual_lathes	Jobbing	150.00	0.200
Amada	Band Saw	band_saws	ERD	20.00	0.100
Herbert7b	Lathe	manual_lathes	Jobbing	120.00	0.300
DeanSmithGrace	Manual Lathe	manual_lathes	Jobbing	20.00	0.300
Mazak0	QT8 CNC barfeed lathe	CNC_lathes	ERS	60.00	0.040
Mazak1	QT8 CNC barfeed lathe	CNC_lathes	ERS	60.00	0.040
Blanchard	Grinder	flat_grinders	ERD	30.00	0.100
Traub0	TND360 Lathe Billet Work	CNC_lathes	ERS	120.00	0.080
Traub1	TNS60 Lathe BarFeed	CNC_lathes	ERD	90.00	0.050
Warner_Swasey	LatheBilletWork	CNC_lathes	ERS	120.00	0.100
Wadkin	NCDrill	CNC_drills	ERS	40.00	0.070
Mazak2	QT15 Robot Loaded Lathe Billet Work	CNC_lathes	ERD	60.00	0.040
Mazak3	QT15 Robot Loaded Lathe Billet Work	CNC_lathes	ERD	40.00	0.000
MazakAGV18a	AGV18 Twin Pallet Machining Centre	machining_centres	ERD	90.00	0.070
MazakAGV18b	AGV18 Twin Pallet Machining Centre	machining_centres	ERD	135.00	0.050
Aumann0	Winding Machine	winding_mc	Coils	30.00	0.100

Warner Electric Machines database (Part 2)

Name	capacity	maxlength	maxwidth	maxbreadth	maxdiameter	rate	btime	tabfeed	rpm
Keithley	70	400.0	400.0	400.0	400.0	95.0	60.00	1000	9500
Hobbing1	70	1000.0	500.0	635.0	635.0	95.0	30.00	00	3500
Hobbing2	70	1000.0	500.0	635.0	635.0	95.0	60.00	00	3500
Ajax	70	1000.0	500.0	635.0	635.0	95.0	20.00	00	3500
Schenk	70	1000.0	500.0	635.0	635.0	90.0	0.00	00	3500
Hardinge	70	300.0	-	-	400.0	90.0	0.00	3400	3500
Herbert7a	70	300.0	-	-	400.0	90.0	0.00	3400	3500
Hi_Ton	70	1000.0	500.0	500.0	500.0	90.0	20.00	00	3500
Branson	70	50.0	50.0	50.0	500.0	100.0	20.00	00	3500
Marsilli	70	50.0	50.0	50.0	500.0	90.0	40.00	00	3500
Eubanks	70	300.0	300.0	300.0	300.0	90.0	30.00	1000	10000
Westminster	70	50.0	50.0	50.0	500.0	90.0	60.00	00	3500
Tapping0	70	200.0	500.0	500.0	500.0	95.0	20.00	1000	3500
Tapping1	70	200.0	500.0	500.0	500.0	95.0	150.00	1000	3500
Tapping2	70	200.0	500.0	500.0	500.0	95.0	150.00	1000	3500
MultiDrill	70	300.0	300.0	300.0	300.0	95.0	30.00	4000	3500
Lapmaster	70	1000.0	500.0	500.0	500.0	95.0	120.00	00	3500
Daniels0	70	1000.0	500.0	500.0	500.0	90.0	40.00	00	3500
Daniels1	70	1000.0	500.0	500.0	500.0	90.0	120.00	00	3500
Pullomax	70	1000.0	500.0	500.0	500.0	95.0	50.00	00	3500
Herbert4	70	100.0	-	-	140.0	95.0	120.00	3500	3500
Herbert2D	70	100.0	-	-	120.0	95.0	150.00	3000	3500
Amada	70	2000.0	200.0	200.0	200.0	95.0	20.00	1000	100
Herbert7b	70	300.0	-	-	200.0	95.0	120.00	3000	3500
DeanSmithGrace	70	300.0	-	-	200.0	95.0	20.00	3000	3500
Mazak0	70	300.0	-	-	160.0	90.0	60.00	3000	7000
Mazak1	70	300.0	-	-	160.0	90.0	60.00	3000	7000
Blanchard	70	400.0	400.0	400.0	400.0	95.0	30.00	1000	20000
Traub0	70	425.0	-	-	600.0	90.0	120.00	8000	4000
Traub1	70	100.0	-	-	60.0	90.0	90.00	4000	6000
Warner&Swasey	70	400.0	-	-	60.0	90.0	120.00	2500	4000
Wadkin	70	200.0	200.0	200.0	200.0	90.0	40.00	4000	1200
Mazak2	70	575.0	-	-	400.0	90.0	60.00	4500	4500
Mazak3	70	575.0	-	-	400.0	90.0	40.00	4500	4500
MazakAG V18a	70	1000.0	500.0	635.0	635.0	90.0	90.00	8000	10000
MazakAG V18b	70	1000.0	500.0	635.0	635.0	90.0	135.00	8000	10000
Aumann0	70	560.0	410.0	400.0	400.0	90.0	30.00	100	6000

Warner Electric Machines database (Part 2)

Name	indextime	feedtime	available	x_coord	y_coord	x_ext	y_ext	power
Keithley	0.000	0.000	True	1.00	20.00	2.00	2.00	1800
Hobbing1	0.000	0.000	True	21.00	21.00	1.00	1.00	0
Hobbing2	0.000	0.000	False	21.00	23.00	1.00	1.00	0
Ajax	5.000	0.250	False	4.00	20.00	2.00	2.00	0
Schenk	0.000	0.000	True	21.00	25.00	1.00	1.00	0
Hardinge	2.000	0.250	False	60.00	48.00	2.00	2.00	7000
Herbert7a	2.000	0.250	False	63.00	48.00	3.00	2.00	7000
Hi_Ton	0.000	0.000	False	50.00	48.00	2.00	2.00	0
Branson	0.000	0.000	False	47.00	48.00	1.00	1.00	0
Marsilli	0.000	0.000	False	70.00	3.00	2.00	2.00	0
Eubanks	0.000	0.000	False	72.50	3.00	1.00	1.00	1800
Westminster	0.000	0.000	False	74.00	3.00	1.00	1.00	0
Tapping0	3.500	0.500	True	37.00	18.00	1.00	1.00	1700
Tapping1	3.500	0.500	False	37.00	16.00	1.00	1.00	1700
Tapping2	3.500	0.500	False	37.00	14.00	1.00	1.00	1700
MultiDrill	2.000	0.250	True	37.00	10.00	1.00	2.00	2800
Lapmaster	0.000	0.000	False	30.00	16.00	3.00	3.00	0
Daniels0	0.000	0.000	True	30.00	24.00	2.00	2.00	0
Daniels1	0.000	0.000	False	35.00	24.00	2.00	2.00	0
Pullomax	0.000	0.000	False	30.00	10.00	2.00	2.00	0
Herbert4	2.500	0.500	False	10.00	10.00	2.00	1.00	0
Herbert2D	2.500	0.500	False	10.00	5.00	2.00	1.00	0
Amada	0.000	0.000	False	40.00	4.00	2.00	2.00	0
Herbert7b	2.500	0.500	False	18.00	18.00	1.00	2.00	0
DeanSmithGrace	5.000	0.250	False	10.00	18.00	2.00	1.00	5000
Mazak0	0.250	0.125	True	24.00	38.00	3.00	10.00	7000
Mazak1	0.250	0.125	False	28.00	40.00	4.00	6.00	7000
Blanchard	0.000	0.000	True	50.00	10.00	2.00	2.00	1800
Traub0	2.500	0.250	True	50.00	13.00	2.00	2.00	31000
Traub1	2.000	0.250	True	50.00	16.00	4.00	4.00	17000
Warner_Swasey	2.500	0.250	False	50.00	22.00	4.00	8.00	9000
Wadkin	1.500	0.100	True	50.00	28.00	4.00	4.00	2000
Mazak2	0.500	0.200	True	60.00	22.00	8.00	4.00	18500
Mazak3	0.500	0.200	False	60.00	10.00	4.00	4.00	18500
MazakAGV18a	0.500	0.200	True	60.00	32.00	6.00	6.00	9000
MazakAGV18b	0.500	0.200	False	60.00	40.00	3.30	3.10	9000
Aumann0	0.000	0.000	True	66.00	3.00	2.00	3.00	100

Factory B Cells Database

name	machine tools	xext	yext	xcoord	ycoord	available	trate
Machine Shop	Swedturn12x4 Swedturn12x2 Swedturn12x2bar Cincinnati25HC Cincinnati20HC1 Swedturn500 Cincinnati20HC2 Cincinnati20HClong CincinnatiT10	40	20	0	0	TRUE	4.5
Fabrication Shop	CincinnatiVC15b Borer CincinnatiVC15a	40	20	0	40	TRUE	5.5

Factory B. Machines database (part 1)

Name	model	type	cell	btime
CincinnatiVC15b	CincinnatiVC15	cnc_drills	FabricationShop	45.0
Swedturn12x4	Swedturn12/4	cnc_lathes	MachineShop	45.0
Swedturn12x2	Swedturn12/2	cnc_lathes	MachineShop	45.0
Swedturn12x2bar	Swedturn12/2	cnc_lathes	MachineShop	45.0
Cincinnati25HC	Cincinnati25HC	cnc_milling_mc	MachineShop	45.0
Borer	-	cnc_boring_mc	FabricationShop	45.0
Cincinnati20HC1	Cincinnati20HC	cnc_milling_mc	MachineShop	45.0
CincinnatiVC15a	CincinnatiVC15	cnc_drills	FabricationShop	45.0
Swedturn500	Swedturn500	cnc_lathes	MachineShop	45.0
Cincinnati20HC2	Cincinnati20HC	cnc_milling_mc	MachineShop	45.0
Cincinnati20HClong	Cincinnati20HC	cnc_milling_mc	MachineShop	45.0
CincinnatiT10	CincinnatiT10	cnc_milling_mc	MachineShop	45.0

Factory B. Machines database (part 2)

Name	x_ext	y_ext	x_coord	y_coord	power	available	capacity
CincinnatiVC15b	2.0	2.0	35.0	5.0	11000	True	70
Swedturn12x4	2.5	4.0	1.0	22.0	35000	True	55
Swedturn12x2	3.5	2.0	4.5	20.5	35000	True	88
Swedturn12x2bar	2.0	5.0	8.0	21.0	35000	False	52
Cincinnati25HC	4.0	4.0	11.0	21.0	7000	True	50
Borer	4.0	4.0	16.0	5.0	5000	False	87
Cincinnati20HC1	4.0	4.0	21.0	21.0	15000	True	33
CincinnatiVC15a	2.0	2.0	30.0	5.0	11000	False	47
Swedturn500	4.0	2.5	3.5	36.0	60000	True	66
Cincinnati20HC2	4.0	4.0	28.0	21.0	15000	False	23
Cincinnati20HClong	6.0	4.0	33.0	21.0	15000	True	46
CincinnatiT10	4.0	5.0	41.0	21.0	7500	True	90

Factory B. Machines database (part 3)

Name	maxlength	maxwidth	maxbreadth	maxdiameter	rate	tabfeed	rpm
CincinnatiVC15b	1000.0	500.0	635.0	-	60.0	3800	6000
Swedturn12x4	1465.0	-	-	20.0	60.0	8000	3000
Swedturn12x2	1465.0	-	-	508.0	60.0	8000	3000
Swedturn12x2bar	1465.0	-	-	508.0	60.0	8000	3000
Cincinnati25HC	500.0	500.0	500.0	-	60.0	3000	3150
Borer	500.0	500.0	500.0	-	60.0	1500	2500
Cincinnati20HC1	800.0	812.0	600.0	-	60.0	1500	2500
CincinnatiVC15a	1000.0	500.0	635.0	-	60.0	3800	6000
Swedturn500	1565.0	-	-	650.0	60.0	9500	3000
Cincinnati20HC2	800.0	812.0	600.0	-	60.0	3000	3150
Cincinnati20HClong	1300.0	812.0	600.0	-	60.0	3000	3150
CincinnatiT10	660.0	660.0	660.0	-	60.0	3800	4000