This book was bound by
Badminton Press
18 Half Croft, Syston, Leicester, LE7 8LD
Telephone: Leicester (0533) 602918.

A CAD ENGINEERING LANGUAGE TO AID MANUFACTURE

by

Nicholas M Hart

A Doctoral Thesis

Submitted in partial fulfilment of the requirements for the award of

PhD of the Loughborough University of Technology

ABSTRACT

The consideration of manufacturing objectives at the design stage is a real possibility given the recent availability of solid modellers within Computer Aided Design (CAD) systems. However, designers and engineers can experience difficulty in communicating with these CAD systems because of the mathematical language which is often used. The ability to capture and transfer design information pertinent to manufacture is also generally rather limited.

This thesis describes the development of a language that uses familiar engineering terminology to specify solid models of engineering components. The descriptive nature of the language naturally incorporates information useful to 'downstream' manufacturing functions; an example of process planning for turning is included.

In order to test the principles proposed a questionnaire was administered to fifty experienced engineers from a wide range of disciplines. The findings of this questionnaire are presented.

ACKNOWLEDGEMENTS

I would particularly like to thank the Science and Engineering Research Council for supporting the research, Dr K Case and Professor R J Sury for their guidance, and the following companies for their help: Rolls-Royce, Baker Perkins, Davy Morris and Brush Switchgear.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF GRAPHS

1 INTRODUCTION

Computer Aided Design (CAD) and Computer Aided Manufacture (CAM) may be considered as two originally distinct areas which are now frequently linked together in a sequential fashion. Unfortunately, the independent evolution of many CAD and CAM systems has created barriers to their integration, often resulting in an incomplete and inefficient communication interface between design and manufacture.

One of the most crucial aspects of an integrated design and manufacturing system is the means to establish and manipulate a computer representation of an engineering component that contains information useful to all stages of the manufacturing process. The extent to which this information is already held within 2D CAD systems is proving insufficient to fully satisfy the requirements of 'downstream' manufacturing needs. The root of a more realistic and comprehensive computer representation of an engineering component is the 3D solid model. Once defined, computer techniques would provide the same facility with the solid model that they currently demonstrate by the ease with which 2D drawings can be stored, retrieved and revised.

There is no doubt that considerable progress has been made in the mathematical modelling of geometric features using solid modelling systems. However, these modellers are not fully developed as engineering tools and omit several essential features, like the ability to incorporate tolerancing information. Consideration must also be given to a host of other factors, especially the requirements of process planning, which is the next vital link in the design to manufacture chain. Process planning is usually the specification of the machine tools, tooling, set-up, technique, machining parameters, materials, job-times, costings, operations and sequences necessary for the manufacture of a component.

The need to design for manufacture is now widely accepted as a necessity if efficient and economic production is to be achieved. Hence production criteria are increasingly represented at an early stage. This makes design the most obvious level to generate the component model, it is also the most desirable from the viewpoint of genuine Computer Integrated Manufacture (CIM), yet it appears that no system at present exploits this potential.

Contemporary research is focused on the development of generative process planning systems, and more recently the application of expert systems. The pursuit of a fully generative process planning system, though attractive, is daunting and to date this goal has not been achieved. The principle obstacles to success are the automatic recognition of machined surfaces from a 3D solid model, the selection of primary processes and optimum manufacturing sequences. It is hoped that these problems can be solved through the use of artificially intelligent programs. Much of the remaining work is potentially algorithmic and does not justify the same approach (typified by the increased usage of 'canned cycles' in numerically controlled machining).

Ironically, interpreting the features of a solid model is a partial reversal of the design process, reconstructing from the finished design the features which the designer has encoded. Moreover, if design for manufacture is practised in all senses, the designer must consider the production processes and their sequence of application in designing a component. Hence if it is profitable to consider such criteria at the design stage then it would seem illogical and inefficient to discard the information only to subsequently synthesize it. This situation is clearly the case if process planning is divorced from design in the manner that the contemporary approach implies. There is also the risk that the design will be treated as an 'absolute' in generating the process plan, whereas changes in the design might be necessary for a truly optimal plan.

The approach described in this thesis differs from much contemporary thought in so far as design and process planning are regarded as indivisable activities. It is proposed that much valuable information can be extracted at the design stage if a different approach is used and that such an approach is intrinsic to the concept of CIM.

The capture of non-geometric information at the design stage poses problems in the design of a CAD system. Information pertinent to process planning is characteristically textual rather than graphical, therefore, new facilities must be available to designers to describe the manufacturing attributes of the design. It will also be necessary to record and present this manufacturing information in a format that is readily useable in process planning and other manufacturing functions. Consequently, considerable emphasis is placed on the investigation and derivation of a new CAD language to fulfil this need.

Before designing such a language numerous factors must be considered, including an investigation of the principle elements linking design to manufacture. The importance of design for manufacture and the nature of modern design practice must be considered. Techniques for rationalizing and managing large volumes of data concerning practices and resources are also important. A key element in linking design to manufacture already mentioned is the representation of engineering components in computer terms. Finally, the relationship with process planning, both present and future, is clearly important. These factors are discussed in more detail in the next chapter.

The recent growth in number and use of computer-aided process planning systems requires more serious analysis than can be attempted in chapter two alone. Therefore, a detailed survey of approaches and systems is undertaken in chapter three, where it is proposed that four types of planning system can be identified: variant, variant-generative, semi-generative and generative. The important differences in the approaches used are discussed, followed by examples of actual process planning systems, with particular attention given to expert systems.

This work establishes the foundations upon which a new approach to integrating design and process planning is proposed in chapter four. The specification for a new CAD language is presented which uses 'familiar engineering language'. Familiar engineering language is not just a language but an approach to design by which important manufacturing information can be collected at the design stage. The techniques for deriving familiar engineering language are described and tested by applying them to the example of turning in chapter five.

The application of familiar engineering language to turning investigates and confirms the technical feasibility of the approach, in software and hardware terms. However, several important practical observations and questions are raised upon which the success of an industrial implementation of familiar engineering language would depend. In order to resolve these practical issues a questionnaire was administered to over fifty experienced engineers from a wide range of occupations and companies. The content and results of the questionnaires are detailed in chapter six.

Although the questionnaire is by no means the definitive test, the evidence of its findings add weight to the practicality of familiar engineering language. The extrapolation of the principles underlying familiar engineering language to encompass other machining process is discussed in the final chapter. In particular, to illustrate the parallelism with other metal cutting processes, the case of cylindrical grinding is reviewed and highlights the potential for wider application of familiar engineering language.

2 THE DESIGN TO MANUFACTURE LINK

Establishing a link between computer aided design and computer aided manufacture, is a high priority for many manufacturing organizations. The benefits obtained so far from computerizing isolated activities in design and manufacturing, have convinced many that computer integrated manufacture is the solution to many of the problems facing western manufacturing industry today.

Fennell (1) suggests if British maufacturing industry is going to survive in anything like a viable form into the next century it has to move as quickly as possible into the age of CIM. This view is also supported by Knill (2) who concludes that the future of America's manufacturing industry lies in the competitive strategy of integrated manufacturing.

CIM is an umbrella term that encompasses almost every possible application of computers within the manufacturing enviroment, from management, through design to production and beyond. The exact nature of CIM is yet to be clarified, however, it is apparent that CIM implies a radical restructuring of traditional manufacturing organizations. Not only will the manufacturing shop floor be further transformed by unmanned machines, robots and automomated conveyancing systems, but also management and technical support. The enormous capital investment in production equipment of this kind requires that it be utilized to the fullest extent. Therefore, with reductions in the manufacturing cycle times as well, management efficiency has to increase dramatically. Inevitably, computers have an increasing role to play in this respect, transforming the 'top-floor' and shop-floor alike.

One aspect which has long been of particular importance in the pursuit of CIM is linking CAD and CAM. Hegland (3,4) champions this point and Halevi (5) argues strongly for an integrated manufacturing approach too, highlighting in particular the interface between design and process planning as an area for attention.

Attention is also focussed on the CAD/CAM link and in particular the roles which design and process planning have in it. Therefore, a background of established philosophy for integrating these activities is now presented.

2.1 Designing for manufacture

Design for manufacture is based on the principle that decisions taken during design, without regard for how the design is to be manufactured, often result in a product which is unnecessarily costly to produce.

In increasingly competitive markets, organizations are finding it more important than ever to minimize manufacturing costs in order to maintain or expand market share. Therefore, the practice of design for manufacture has arisen not out of theory, but in answer to a genuine realization of the economic implications of design on manufacturing efficiency.

Considerable potential has been shown throughout the engineering industry for cost avoidance by designing for manufacture. Cook et al (6) cite the example of an analysis undertaken at Rolls Royce, where a cost reduction team examined 2000 drawings and found that of all the possible ways in which costs could be avoided, 80 per cent of them were design related.

The point is more generally stated by the British Standards Institution in the Manual of British Standards in Engineering Drawing and Design (7), from which the following paragraph addressing the influence of design on economic production is taken:-

> Product costs only originate in design and the designer has a prime responsibility to ensure that the product gives optimum value for money. The irreducible cost of the product is determined by the designer. A poor design may commit a company

to many years manufacture of a design which would have been more profitable had the decision been better conceived. Cost is as much an attribute of the design specification as is performance, appearance, reliability, life, safety, etc., and is an essential factor to be satisfied by the optimum design solution.

In order to understand how to integrate production considerations into design, it is first of all necessary to examine contemporary design practice in more detail. Although there are undoubtedly many different approaches to engineering design, in general it can be viewed as a three stage process:-

1. Design specification
2. Conceptual design
3. Detail design

2.1.1 Design specification

The corporate strategy of most manufacturing organizations is to market products or processes profitably. Any new product must form part of such an overall strategy. Consequently, the starting point for most design specifications is the results of market evaluation and research. Building on the results of market research, design specification is a process of determining the objectives and constraints for all subsequent design activity.

To construct a complete design specification numerous factors must be considered, Cook et al (6) identify twenty-eight, spanning across the whole spectrum of manufacturing organization. Some of the key factors can be summarized as follows:-

market demand	capacity
processes	materials
ergonomics	performance
cost	quality
quantity	maintenance
patents	aesthetics
longevity	proving
size	packaging
transportation	law

From the manufacturing point of view, the choice of materials, processes, quantities and quality, have obvious effects on production costs. For example, if a design cannot be manufactured by existing processes then additional capital investment will be necessary or alternatively the work must be contracted out. Similarly, the choice of workpiece material affects the choice of tool materials, rake angles or grit values, feeds and speeds. Quantity has implications for the manner of production (job, batch, line), whereas high quality may imply additional finishing processes or higher scrap rates.

During design specification it is beneficial to precisely define the framework within which subsequent design work must take place, in order to focus the attention of the design team. However, care has to be taken not to over specify and therefore predetermine the outcome of the design effort.

2.1.2 Conceptual design

Once the design specification for a product has been completed, the next stage in the design process is often called conceptual design. The purpose of conceptual design is to generate concepts that fulfil the objectives set down in the design specification, then by systematically analysing the viable alternatives to identify the most suitable solution.

Of the many techniques developed to aid or systematize conceptual design, Turner (8) has identified sixteen different approaches. These range from basic techniques like the use of biological analogies, attribute listing and brainstorming, to more esoteric techniques like lateral and parallel thinking.

A review of design methods has been undertaken by Pugh and Smith (9), based on experience in teaching and consultancy work over a period of fourteen years. The conclusion of this work was that the methods which proved most useful in practice were invariably the simplest ones: namely analogy, inversion, attribute listing and T-charts. Of which most are probably practised unconciously by designers anyway. The single most productive design technique was identified as informal group discussion.

As with the preliminary activity of design specification, the generation of concepts and selection of the optimum one, is more efficiently performed by group working. When the members of the design group are multidisciplinary, comprising of engineering designers, industrial designers, ergonomists, mechanical engineers, production engineers, electrical engineers and so forth, the resulting design concept is more likely to exhibit the variety of features which distinguish a successful product.

2.1.3 Detail design

Traditionally, the detailed engineering drawing has been the primary means of communication between the design and production departments within an organization. It is these drawings that form the tangible conclusion of design. While it is important to be correct in the specification and conception of design, it is equally and sometimes more important to detail the design properly because it embodies the ultimate result of all the design work.

Detail design is largely concerned with attributing manufacturing tolerances, specifying materials and other design details. All of these affect the subsequent choice of production processes and have critical influence on manufacturing costs.

For example, if three identical components are turned one each from mild steel, aluminium and brass, the respective raw material costs are approximatlely doubled in each case (see fig 2.1). That is to say that the stock bar for the aluminium component would be approximately twice as expensive as for the mild steel component, and half as expensive as for the brass component.

While the relative costs of different raw materials are easily identified, the relative costs of different tolerances are less obvious but significant. For instance, if the same tolerance is applied to two components having identical internal diameters, where one component can be revolved and other cannot, the internal diameter on the revolvable component will be between approximately three times cheaper to manufacture (see table 2.1). Unless the processes by which these two components must be produced are

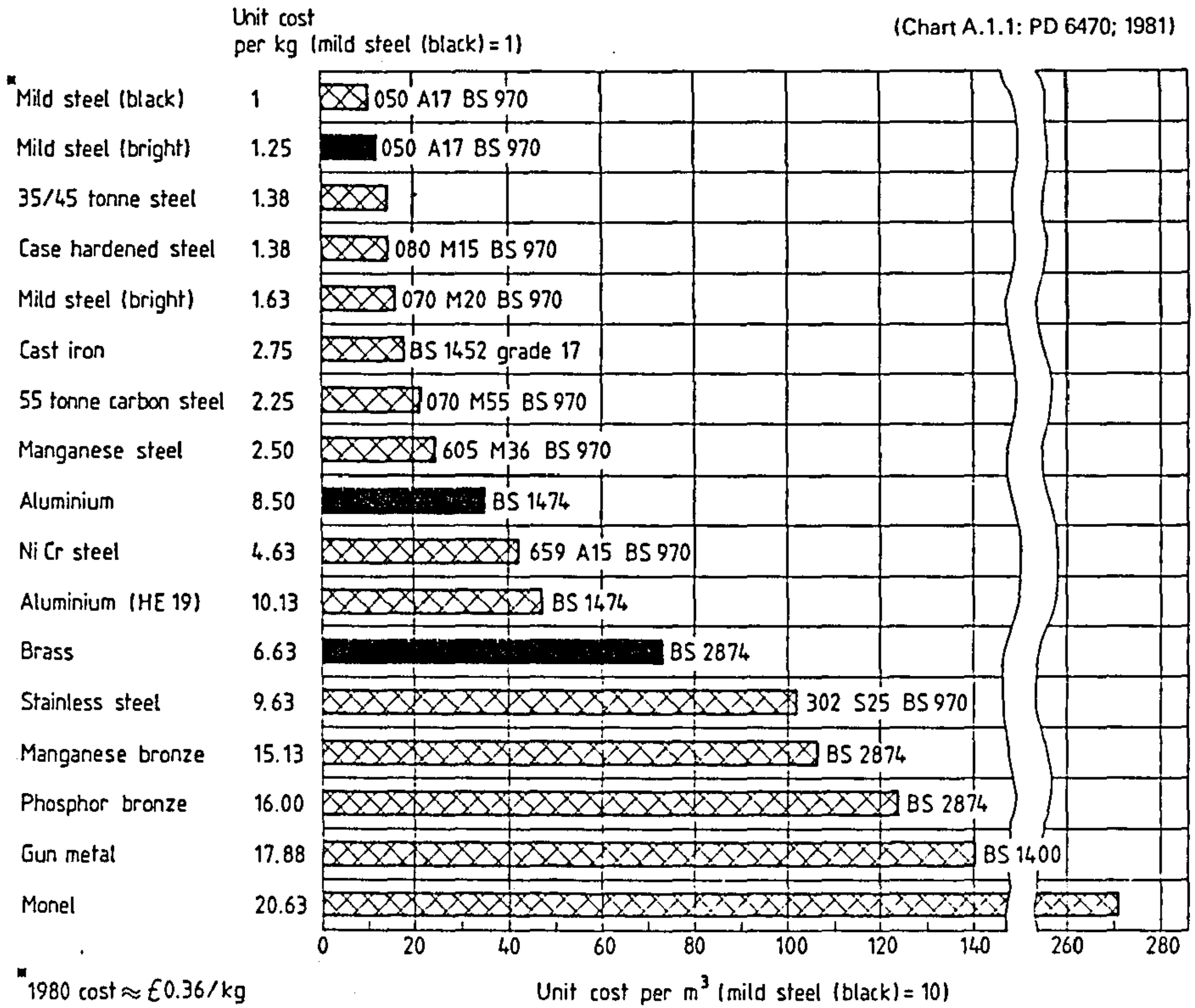


Fig 2.1 Material price comparison of ferrous and non-ferrous bars: British Standards Institution (7)

Feature	Tolerance on element Full Indicator Movement, mm							
	0.30	0.15	0.075	0.04	0.02	0.01	0.005	0.003
Face of a bore *	0.56	0.70	1.02	2.00	4.43	5.56	7.06	
Gear centre *	0.18	0.26	0.37	1.98	2.86	3.46	5.48	
Internal diameter *	0.57	0.76	1.07	1.51	2.31	3.32	5.05	
Plain face on frame	0.15	0.31	0.38	0.86	1.16	2.59	3.84	
Face of a bore	0.33	0.38	0.44	0.55	0.68	1.90	4.32	6.52
Internal diameter	0.19	0.24	0.29	0.39	0.50	1.07	2.37	3.67
External diameter	0.06	0.06	0.07	0.20	0.21	0.24	0.47	0.70

* Non-revolvable

Table 2.1 The relative manufacturing costs for different features and tolerances: British Standards Institution (7) considered, the reason for the difference in manufacturing costs is not immediately obvious. In one case a turning operation would be implied, whereas in the non-revolvable case a jig-boring operation would be implied. By considering the production processes necessary to obtain the specified tolerances, the relative costs become more apparent.

Another example is a comparison of the costs of applying two different tolerances to the same feature of a component. If an external diameter is to be produced on a component within Ø.075mm and Ø.04mm respectively, the cost of producing the component to the tighter tolerance is likely to be three times higher (see table 2.2). Again the reasoning is clear if the manufacturing implications are considered. A Ø.075mm tolerance can be achieved economically on a turret lathe, whereas a Ø.04mm tolerance almost certainly implies the use of a centre lathe (see fig 2.2).

Of course in certain cases it will be essential to specify small tolerances to obtain the necessary performance. In other instances the choice of arbitrary values is very costly, as the simple examples (above) taken from British Standards (7) show. Clearly there are major financial incentives to design for manufacture. Moreover, the appreciation of production costs in design is significantly improved by considering the actual manufacturing processes

Existing tolerance band FIM, mm	Proposed tolerance Full Indicator Movement, mm							
	0.30	0.15	0.075	0.04	0.02	0.01	0.005	0.003
0.30		1.00	1.20	3.29	3.53	4.01	7.87	11.6
0.15	1.00		1.20	3.29	3.53	4.01	7.87	11.6
0.075	0.83	0.83		2.75	2.94	3.34	6.56	9.69
0.04	0.30	0.30	0.36		1.07	1.22	2.39	3.53
0.02	0.28	0.28	0.34	0.93		1.14	2.23	3.29
0.01	0.25	0.25	0.30	0.82	0.88		1.96	2.90
0.005	0.13	0.13	0.15	0.42	0.45	0.51		1.48
0.0025	0.09	0.09	0.10	0.28	0.30	0.35	0.68	
% scrap expected				1	2	4	7	12

Table 2.2 The relative manufacturing costs of an external diameter that can be revolved: British Standards Institution (7)

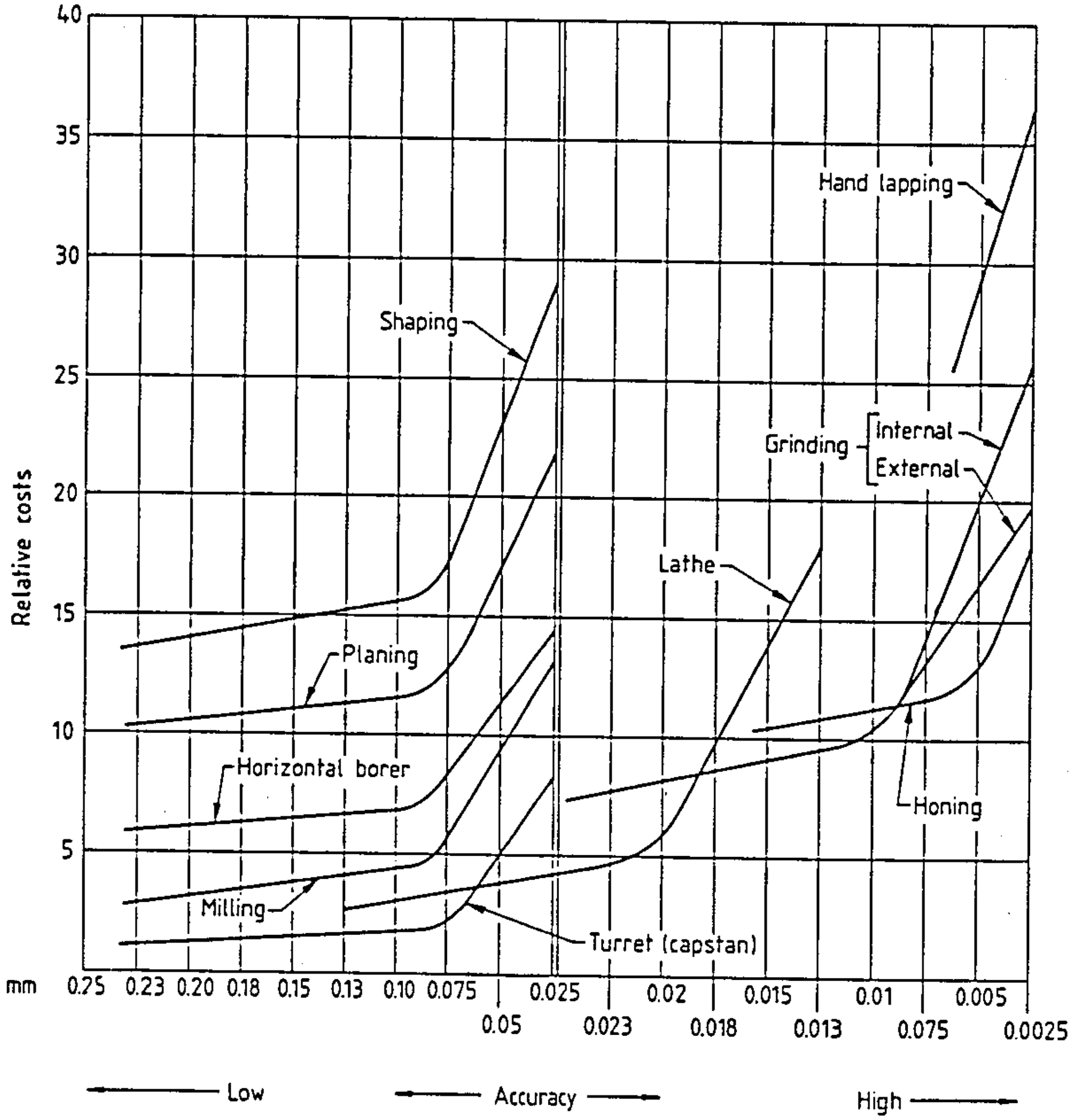


Fig 2.2 Cost of various machine and hand processes for achieving set tolerances: British Standards Institution (7)

and sequences implied. These considerations are as important in a computerized interface between design and manufacture, as they are in the manual one.

Many other factors can adversely effect production costs, among them is surface finish requirements (see table 2.3). Less quantifiably, are non-consideration of basic production practices (leading to components requiring excessive numbers of processes, operations and set-ups) and the use of non-standard components and sizes. Standardization is particularly important for cost effective design.

2.2 Standardization, classification and coding

Standardization, classification and coding are three major tools for creating structure and order in manufacturing organizations. Standardization has been fundamental to the engineering industry virtually since the industrial revolution. Evident at many different levels, internationally (ISO), nationally (BSI), industrially and by company, it has been broadly defined (7) as:-

> The discipline of using the minimum number of parts for the maximum number of purposes, produced by the most economical manufacturing processes, of the appropriate quality to give reliable and acceptable

Surface texture		Ra mm-3	Roughness grade number	Relative cost
Machined	Very rough (Very coarse feed)	50	N12	1
	Rough	25	N11	3
	Semi-rough	12.5	N10	6
	Medium	6.3	N9	9
	Semi-fine	3.2	N8	13
	Fine	1.6	N7	18
Ground	Coarse	0.8	N6	20
	Medium	0.4	N5	30
	Fine	0.2	N4	35
Lapped	Super fine	0.1	N3	40

Table 2.3 The relationship between surface finish and manufacturing costs: British Standards Institution (7)

performance at minimum (whole life) cost.

That standardization must be practised in design is an accepted principle, therefore it will also be an essential ingredient in an integrated CAD/CAM system. If the extent of standardization has hitherto depended on the ability of designers to recollect and research standards, by incorporating information on standards into computer systems, the potential exists for the designer to be more reliably informed. The problems of computerizing such volumes of data demand that a highly structured approach is used.

A technique that has proved successful in the management of large volumes of interelated information is classification and coding. Hyde (10) defines classification as a technique to organize any related data in a logical and systematic order that groups like things together.

Whereupon he amplifies that classification on its own is largely ineffective without an efficient means to handle and process the data, namely by coding it using symbols (usually alphabetic and/or numeric characters). Perhaps the best example of classification and coding is the Dewey decimal system, used in libraries all over the world (see fig 2.3). The same principles have also been applied to great effect in the manufacturing industry, as for instance in Group Technology.

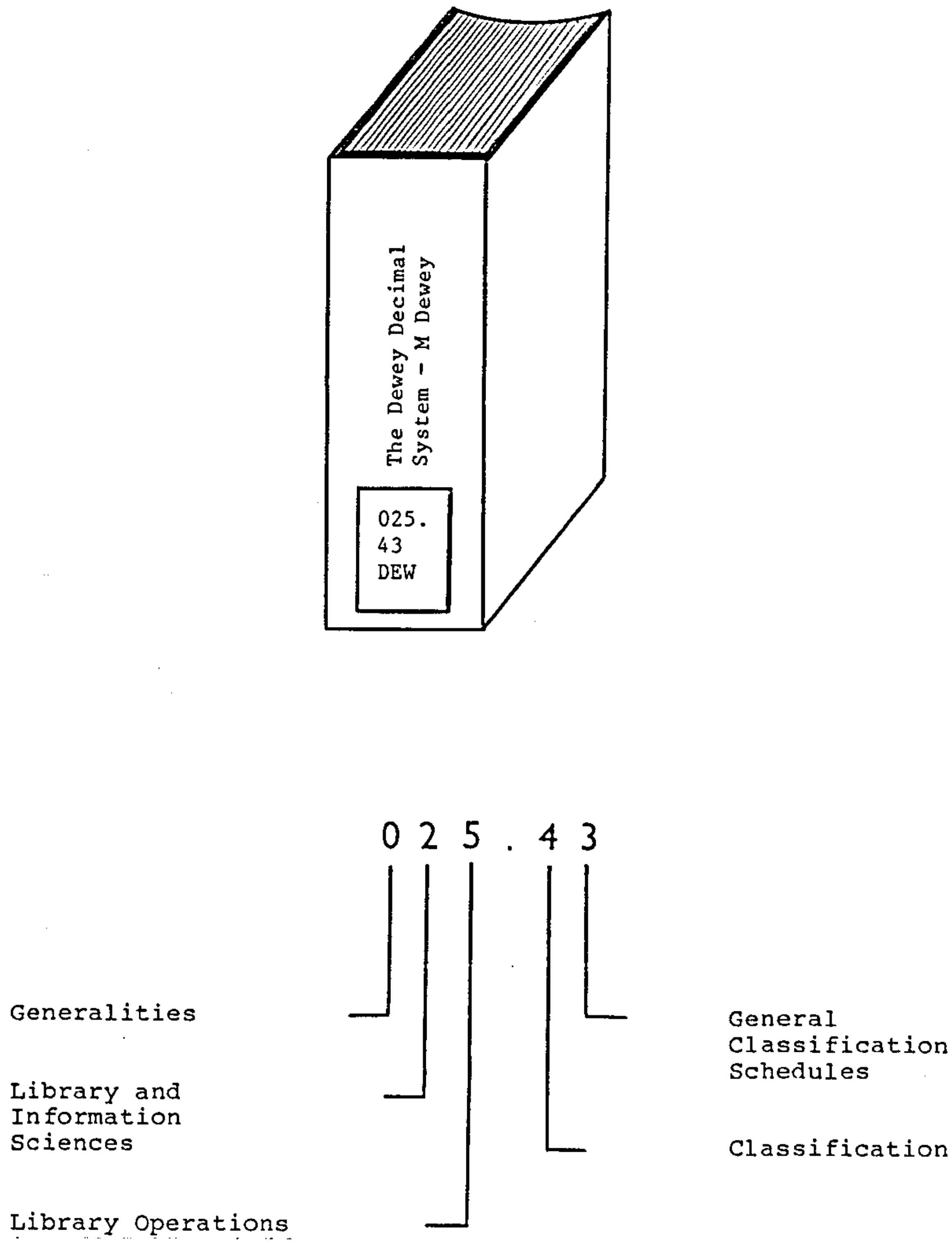


Fig 2.3 An example of the Dewey decimal classification system

Group Technology (GT) is both a manufacturing philosophy and an organizational principle. It is basically, the concept that manufacturing methods are often common to groups of otherwise dissimilar components. Therefore, by grouping together processes, physically into manufacturing cells (the ideal case) or notionally if in process layouts, production efficiency is improved. Typically, benefits include reduced manufacturing lead times, fewer set-ups and less work in progress inventory as a result of smaller batch quantities.

The successful implementation of GT usually requires that parts and materials are classified and coded, in order that candidate part families/groupings can be selected by the analysis of design shapes, engineering features and manufacturing methods. Although it is recognised that other techniques exist, for example the technique of production flow analysis as described by Burbidge (11).

Classification and coding while often synonymous with GT has general application in the organization of manufacture, Burbidge summarizes that the aim of classification and coding is to provide a rapid and efficient method of retrieval for decision making.

Computerization is pursued for similar objectives, and it follows that the application of classification and coding as a methodology for structuring manufacturing information, may also be a prerequsite in unifying CAD and CAM. Which Lardner (12) defines as a series of data processing operations in the final analysis. The creation and management of large databases for design and manufacture, is simplified if the information is well structured.

2.2.1 Techniques for classification and coding

No completely universal classification and coding system for manufacturing information exists. This is undoubtedly due to the complexity of the manufacturing enviroment and the multiplicity of uses which must be served.

For example, a system suitable for classifying and coding drawings is frequently different from one appropriate for classifying and coding information for production. Shaffer (13), identifies these as the two major uses of classification and coding in the engineering industry, adding that the use of seperate design orientated and production orientated systems is common.

One of the more significant differences between different industrial classification and coding systems is the manner in which they are coded. Three major types can be identified:-

(a) Monocodes

(b) Polycodes

(c) Mixed or hybrid codes

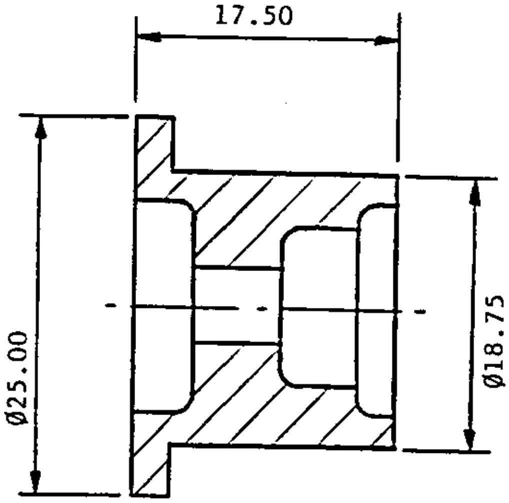
Of which, monocodes are frequently considered to be design orientated, polycodes to be production orientated, whereas mixed codes are a compromise format suitable for both applications.

(a) Monocodes

The monocode is a hierachical code in which each element amplifies the information given in the previous element. They are difficult to construct but the result is compact and detailed because of its interdependant nature. Hence they are better suited to permanent forms of information.

A disadvantage of the monocode results from its sequential nature in so far as the meaning of monocodes can be relatively opaque. Typically, monocodes are more difficult to computerize than others in terms of the efficiency with which databases can be accessed.

An everyday example of a monocode, already mentioned, is the Dewey decimal system. Industrial examples include the Brisch code (11), developed to help simplify design information referencing and retrieval. An example of a component coded



Surname or family name — Christian name

Code number 3 3 2 6 5 – 2 04

- 3: Single piece part
- 3: Metallic, round multi-diameter
- 2: Straight c/line max. section at one end, without thread, with through-going c/hole
- 6: Round plain multi-diameter c/hole, maximum outside dia. up to and including 4Ø.ØØ
- 5: Short, two external diameters only, minimum inside diameter between ends
- 2: Length up to and including 2Ø.ØØ
- 04: The fourth component of this type

Fig 2.4 An example of the Brisch coding system:

Burbidge (11)

using the Brisch system is shown in fig 2.4.

(b) Polycodes

Polycodes are feature orientated and independent in construction. Individual elements of a polycode have seperate meaning, thus they can be easier to construct and modify. Although well suited to computerization, in order to convey a useful amount of detail polycodes become rather large and therefore less manageable by manual methods.

(c) Mixed or hybrid codes

A mixed code generally consist of part monocode and part polycode in order to obtain benefits of both approaches and provide a more universal coding system. In common with most compromise solutions, universal codes can prove less than optimal in specific cases.

Many industrially used codes are of mixed format. The Opitz system (30) uses a mixed code to describe engineering components (see fig 2.5), another example of the mixed code is MICLASS (32,33), developed by the Organization for Industrial Research (OIR), which has a capacity for up to thirty digits. Such mixed codes have proved to be useful instruments for integrating process planning and manufacturing, a case study using MICLASS is described by

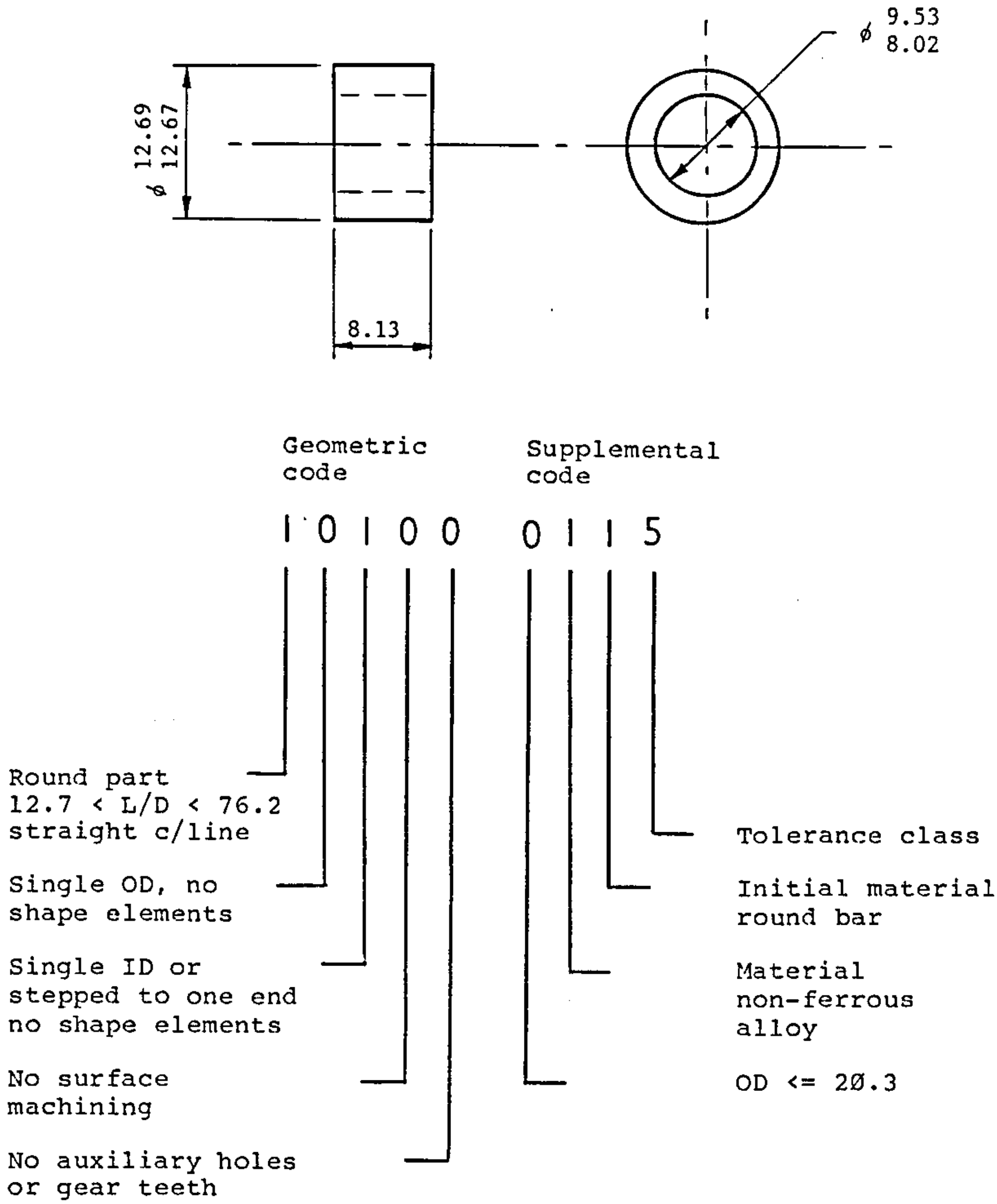


Fig 2.5 Example of the Opitz coding system

Shaffer (14).

Classification and coding systems have wide application in manufacturing organizations, from design standardization to tool and machine rationalization. The structuring of manufacturing data is a prerequisite for computerization, in this respect they can be of considerable benefit. Another area important to the integration of design and manufacturing information is the description of product geometry, in terms that are useful for manufacture as well as design purposes.

2.3 From engineering drawing to Product model

Over the past few years there has been a revolution in the preparation and handling of geometric data. The advent of CAD has resulted in major transformations in design and draughting practice for many organizations. To date the impact of CAD has largely been limited to the implementation of two dimensional (2D) computer draughting systems. Although of immense benefit, subsequent research has demonstrated a further potential for revolutionising design through the use of three dimensional (3D) computer geometric modelling systems.

The 2D drawing, based on orthographic projection using either first angle (European) or third angle (American) has long been a cornerstone of the engineering industry. It is the regulated medium for representing engineering designs to a consistent standard. Nevertheless, the 2D drawing is only a means of encoding information about a 3D solid object. The use of symbols and lines to represent physical properties and real solids is highly artificial, depending on the engineer to form a mental image of the 'real' object. Interpreting a 2D drawing and visualizing the 3D object is not always easy even for the trained engineer.

As the primary means of communicating design information for subsequent use by production engineers, the 2D drawing has also created an artificial engineering interface between design and manufacture. The ability to design a component in 2D is all too frequenlty found to be different from the reality of manufacturing it. The root of a more realistic and comprehensive interface lies in the 3D solid model.

There is nothing particularly novel about 3D modelling, the technique has always been practised. In the aerospace and automobile industries, scale models have long been used for the ultimate visualization and proving of designs. However, the manual production of such models, which is both time consuming and costly, is used by necessity as the last resort. By comparison, computer 3D models are relatively fast and inexpensive to produce.

2.3.1 3D computer solid modelling

Computer solid modelling is relatively complex and a full explanation of the relevant mathematical issues is given by Requicha (15), while a much simplified overview is given here.

Many authors have expounded the virtues of solid modelling in uniting design and manufacturing considerations into a single electronic product description. Carey and Pennington (16) suggest that a geometric modelling system having multiple representations with dimension, tolerances and machining information would facilitate the selection of appropriate manufacturing processes and tooling to produce the specified geometry and surface finish. They further propose that with minimal human intervention a prototype system would give process planned NC code.

The main advantages of using solid models instead of 2D drawings can be summarized as follows:-

(1) Provides a simulated view of an object almost as good as a real prototype.

(2) Enables calculation of mass properties and other engineering characteristics.

(3) Has the potential to integrate design and manufacturing considerations into a single product description.

(4) Is a possible basis for automatically generating and verifying NC code.

(5) The model is mathematically unambiguous.

In any model, there are limitations to the accuracy and degree of detail that can be incorporated. In the case of computer solid models, reality is also represented in an idealized form, for example the engineering considerations of tolerance and surface finish are not presently accommodated. A simple definition of the scope of present solid models has been given by Jared (17) as a representation that allows all points in space to be classified as inside, outside or on the surface of an object. While this description may be mathematically succinct, there are important gaps in the information content of the solid model for engineering purposes. These gaps have to be filled before solid modellers can be employed in an engineering design system.

2.3.2 Representations for solid modelling

In approximately fifteen years of research into solid modelling systems, six identifiable methods of implementing the solid model representation have resulted, described among others by Jared (17) and Sabin (18):-

(a) Constructive solid geometry
(b) Boundary representation
(c) Spatial occupancy enumeration
(d) Sweeping
(e) Cell decompostion
(f) Disjoint primitive instancing

(a) Constructive solid geometry (CSG)

The fundamental principle underlying CSG is that solid objects can be considered as a set of points in space which can be combined using the set (boolean) operators. There is also a theoretical maximum set of geometric primitive volumes from which all other objects can be derived. These consist of the block, cylinder, cone, sphere and torus; often considered as the geometric 'building bricks' (see fig 2.6).

Primitives of any size can be created by varying their characteristic dimensions, then using boolean operations as shown in Fig 2.7 it is possible to add (union), subtract (difference) and intersect these to form a more complex object as in fig 2.8. An object or primitive can also be located anywhere in 3D space using translation and rotation operations. The diversity of construction techniques allowed within CSG means that there are usually many different ways to model a given solid.

While CSG representations are extremely compact, the process of interogating the CSG model, as in updating displays, requires considerable computation. Another disadvantage is an inherent difficulty with certain types of curved surfaces, for example as might be encountered in modelling castings. By contrast, a high level of compatability with machined components is evident, as model construction can emulate

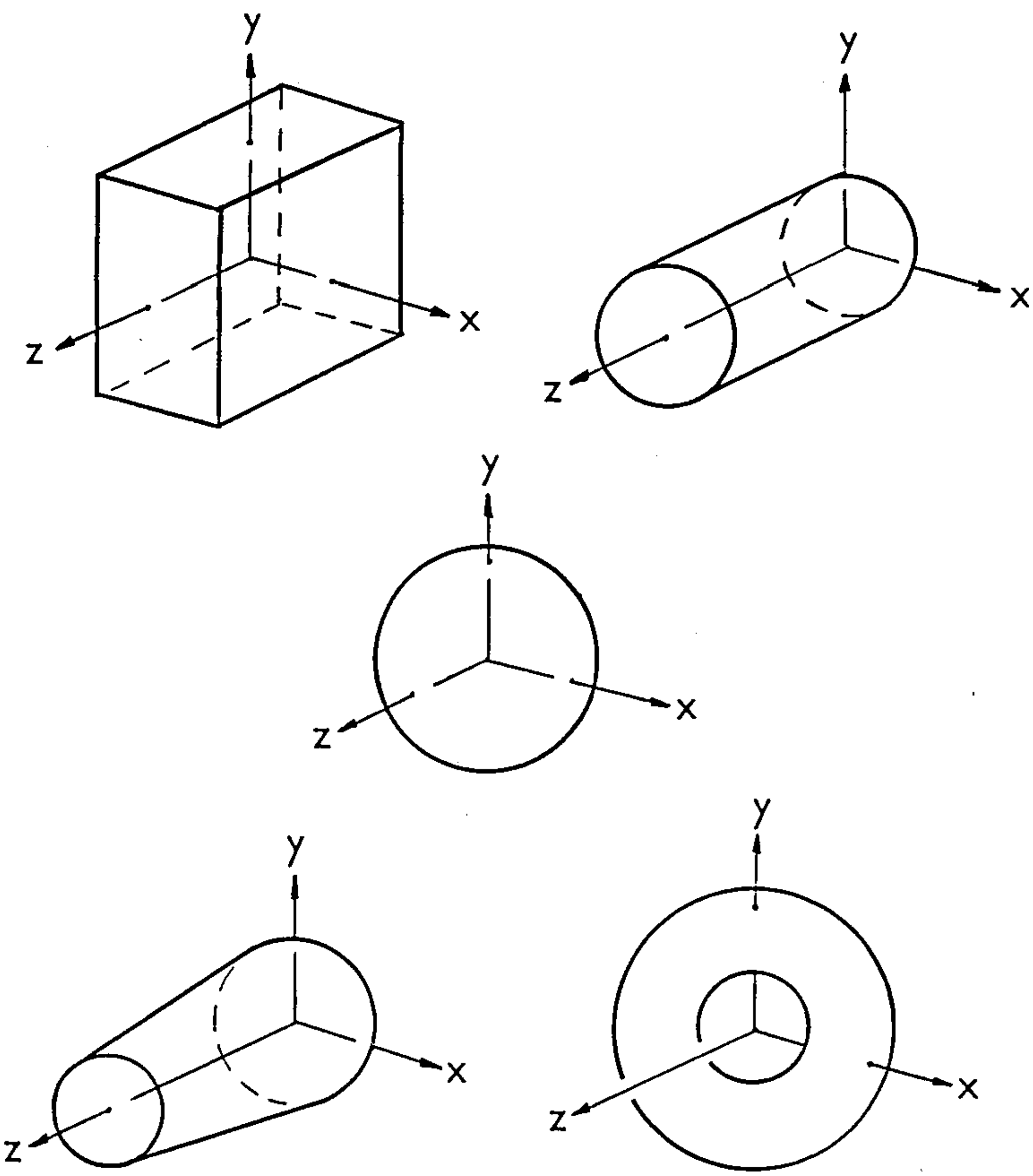


Fig 2.6 Solid modeller 'geometric building bricks'

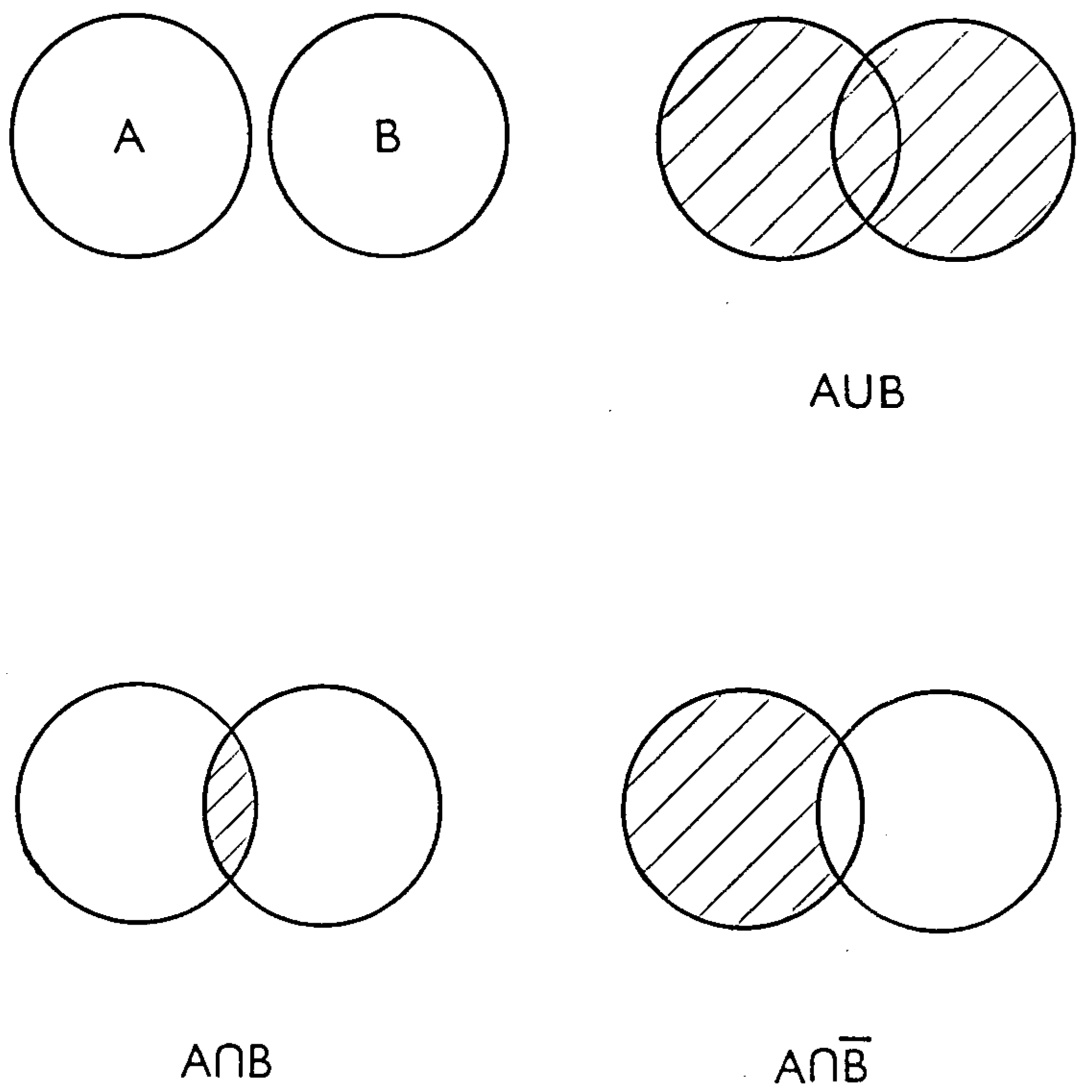


Fig 2.7 The boolean (set) operators

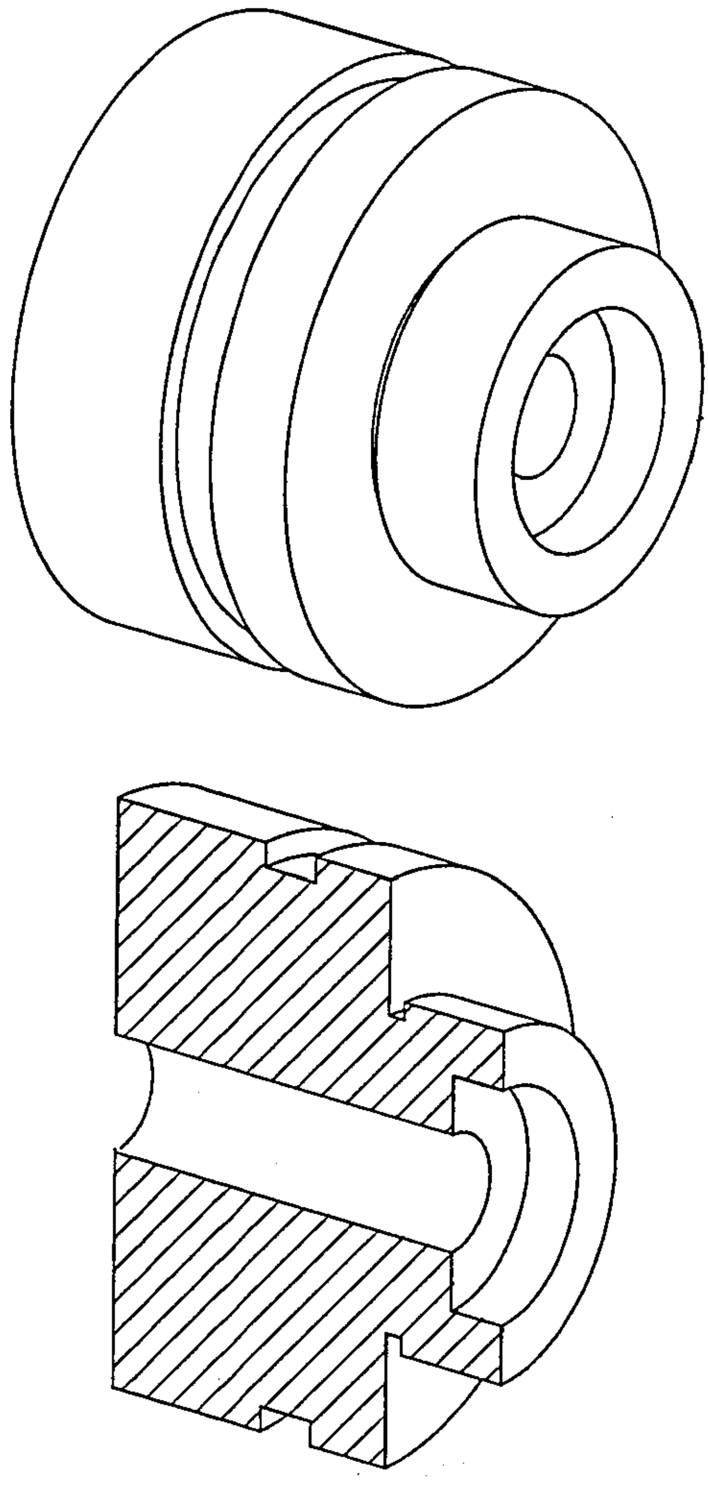


Fig 2.8 Example of a turned component modelled in BOXER

actual machining processes (a more detailed treatment is undertaken in chapter 6).

(b) Boundary representation

Boundary representations can be considered as topological descriptions of solid objects. The object is represented in terms of its bounding faces, edges and vertices (see fig 2.9). In comprehensively noting the interrelationship of these topological features, a boundary representation becomes quite large, even so interrogation is usually faster than for other representations.

A significant advantage of the boundary representation is the ability to represent a wide range of surface types and hence model irregular objects with comparative ease. A disadvantage of boundary representations is that they are not implicitly mathematically guaranteed solids. It is possible for users to create invalid solids if a rigorous user interface is not employed, Woodwark (19) cites the example of pulling a vertex through a face to create a nonsensical object.

(c) Spatial occupancy enumeration

This form of solid representation is achieved by reducing the volume of space encompassing an object into cubes, whereupon

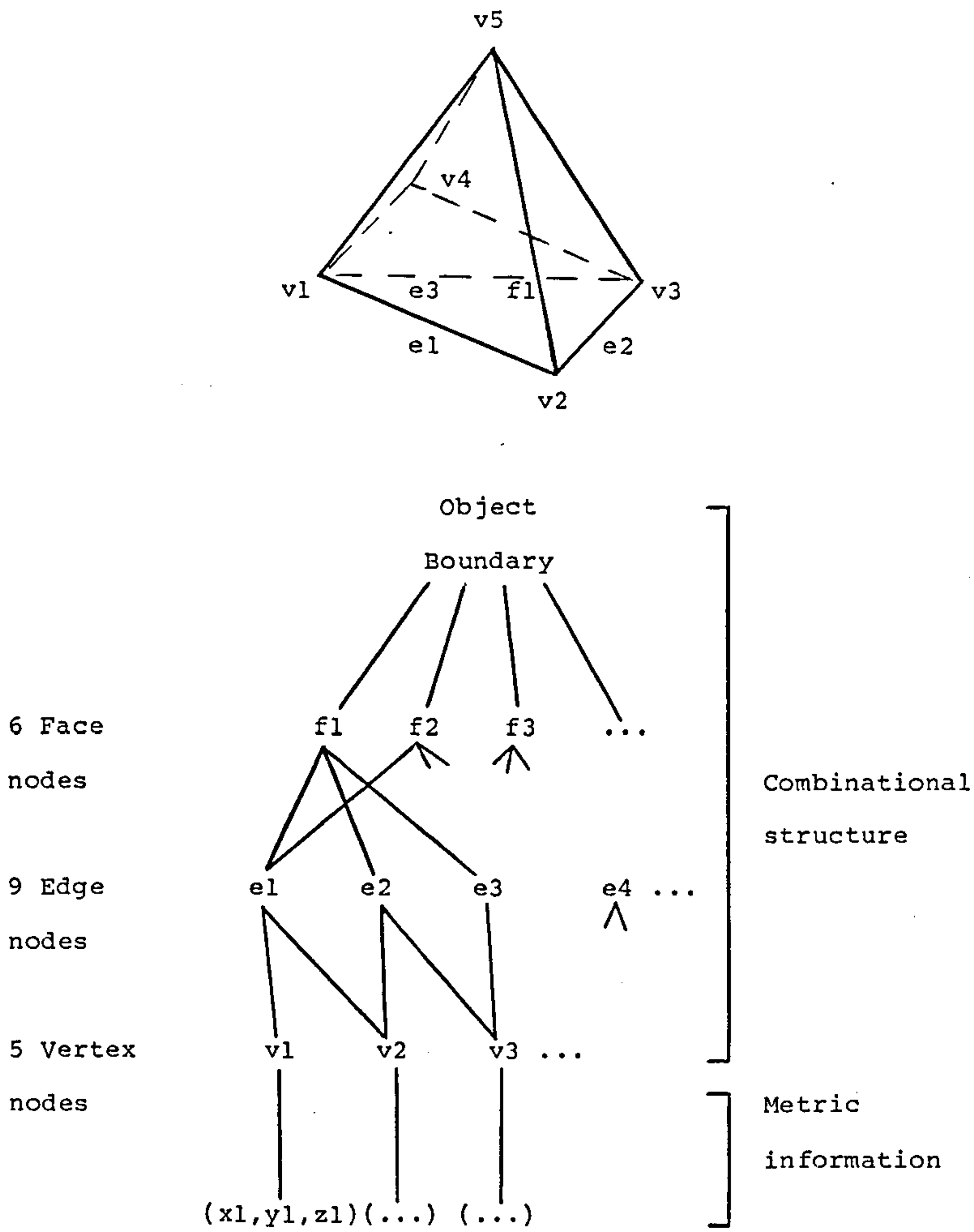


Fig 2.9 A boundary representation for a rectangular pyramid: Requicha (16)

all the cubes that are occupied by the object are enumerated (see fig 2.10). The technique is well suited to the computation of mass properties, although the results are less accurate for objects with curved surfaces. To obtain a suitable degree of accuracy in this case, cube sizes must be reduced to the point where impractical numbers of cubes are involved, from the point of view of computer memory.

A refinement of the technique, the Octree model (20), enables partially occupied cubes to be identified and further sub-divided (recursively), thus reducing the number necessary to attain a reasonable degree of accuracy (see fig 2.11).

(d) Cell decomposition

Cell decomposition can be considered as a more general form of spacial enumeracy occupation, where instead of cubes, an object is decomposed into polyhedra. The polyhedra or 'cells', can be used in the computation of mass properties or as the starting point for volumetric mesh generation in finite element analysis.

(e) Sweeping

Solid objects can be represented as the result of sweeping curves or surfaces, which produce more surfaces and volumes respectively, as in fig 2.12. Although an effective and

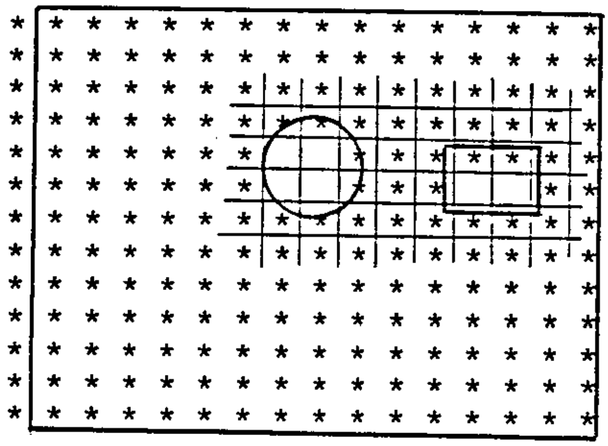


Fig 2.10 Spatial occupancy example: Jared (18)

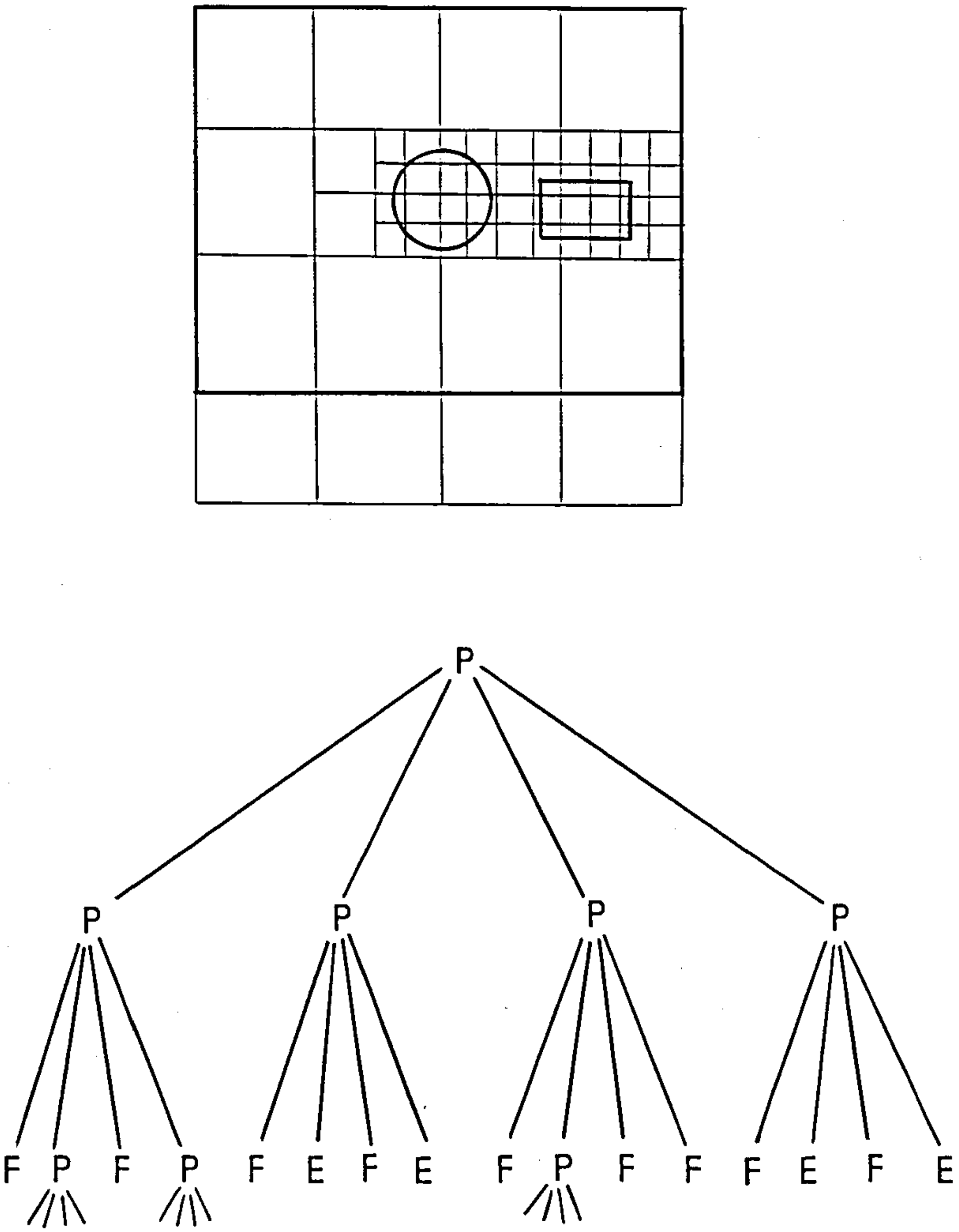


Fig 2.11 Octree decomposition example: Jared (18)

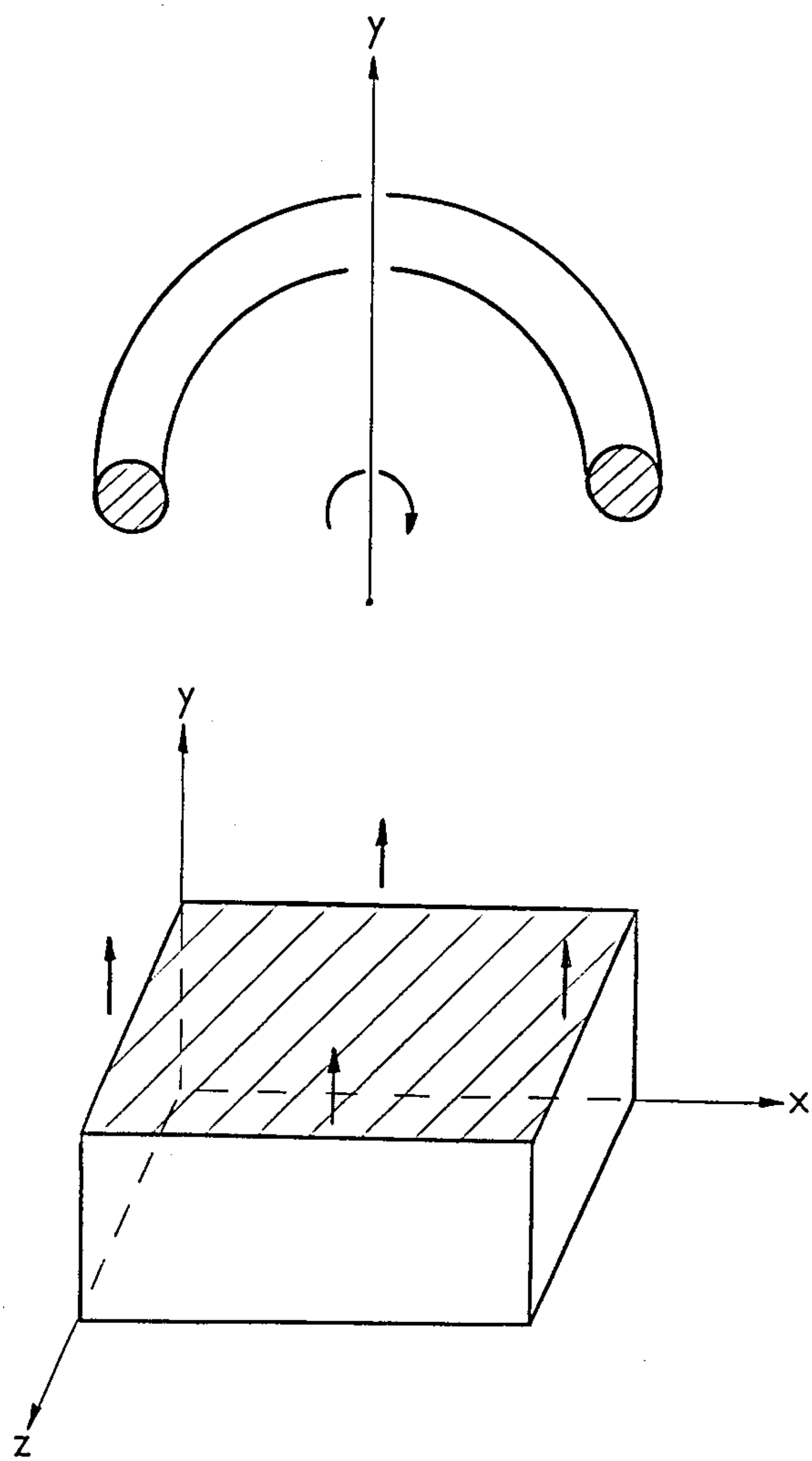


Fig 2.12 Sweeping examples - rotating a disc and lifting a plate

practical technique in certain cases for producing complex solids, the result is not easliy converted to other representations.

(f) Disjoint primitive instancing

A disjoint primitive instancing system contains a predefined set of geometric shapes, similar to CSG primitives, that can only be joined together without overlapping. Although this is relatively fast in computational terms and results in models with a high degree of mathematical integrity, it is limited to specific applications where standard units are to be assembled. Sabin (18) cites the example of representing refineries.

2.3.3 Contemporary solid modelling systems

Many contemporary systems use a multirepresentational approach, converting information internally between several representational formats (outlined earlier). For example, boundary representations are convenient for producing graphic displays, cell decomposition can facilitate mass property calculations, constructive solid geometry is a compact and mathematically guaranteed format for describing solid objects, whereas sweeping can very often be the only practical means of modelling many 'real world' solid objects.

Several authors have reviewed the range of contemporary solid modelling systems, including Pratt (21) and Pipes (22). Notable examples among the increasing number of solid modelling systems are Noname (23), Medusa (24), BUILD (25) and COMPAC (26,27).

Noname was developed at Leeds University, though undoubtedly influenced by earlier collaborative experience with Rochester University during the development of PADL-1/2 (28) solid modellers. A commercial counterpart of Noname, marketed by PAFEC, is called BOXER. Both are CSG based systems, whereby primitive volumes including blocks, cylinders, cones, spheres and the torus can be combined using the boolean operators union, difference and intersection to construct models of solid objects.

Medusa was developed by Cambridge Interactive Systems. Pratt (20) observes that Medusa is different from more conventional solid modelling systems, because information is stored primarily as representations of 2D drawings. Sufficient additional information is associated with the drawings to generate a 3D faceted model whenever it is required.

BUILD is a result of work at Cambridge University, although connections are also apparent with the commercial product Romulus, marketed by Shape Data. BUILD is a boundary representation system driven by a variety of constructional techniques including boolean operations on primitive volumes, the use of sweep (translational) or swing (rotational) operations, as well as localized 'tweaking'.

COMPAC is a solid modelling system developed at the University of Berlin, which uses a combination of NC programing-like language (EXAPT2), solid primitives (similar to Noname) and sweeping contours. Consequently, it is able to model a wide range of engineering components including non-rotational, rotational and sheet metal objects.

2.3.4 Engineering considerations in solid modelling

While the relative mathematical virtues of these different approaches are clearly important, the engineer may be more concerned with the features that are most apparent to him. For example the type of components that can be modelled, the ease with which this can be accomplished and the ultimate usefulness of the result. As between mathematicians, there is some variation between engineers as to the utility of different solid modelling techniques. For instance, engineers working with metal cutting processes encounter different geometric features to those found in casting, forming or fabrication processes.

Depending on the application, the importance of such real world attributes as tolerance is variable. For example in visualising a conceptual design or for the purpose of creating pictorial documentation, tolerance is irrelevant because it cannot be seen. Whereas the construction of a solid model useful in manufacturing requires that tolerances are present because it is impossible to manufacture without them.

Before 3D solid modelling systems can achieve wide acceptance in industry, considerable revision of contemporary practice and attitudes will be necessary. The nature of the 3D solid model is very different from the 2D drawing. To be a credible alternative to the 2D drawing, more information must be included in the solid model, therefore it is clear there is still some way to go in developing solid modellers.

2.4 Computer aided process planning

Once the product design has been finalized and detailed, the next major function in the design to manufacture link is production engineering, which can be defined as the aspect of manufacture concerned with the methods of translating designs into finished products through the most efficient use of production resources. As with design, there is a considerable variation between organizations in the way production engineering is undertaken.

Typically, production engineering involves the following tasks:-

Process planning	Production control
Work study	Costing and Estimating
Quality control	Production management
NC programing	

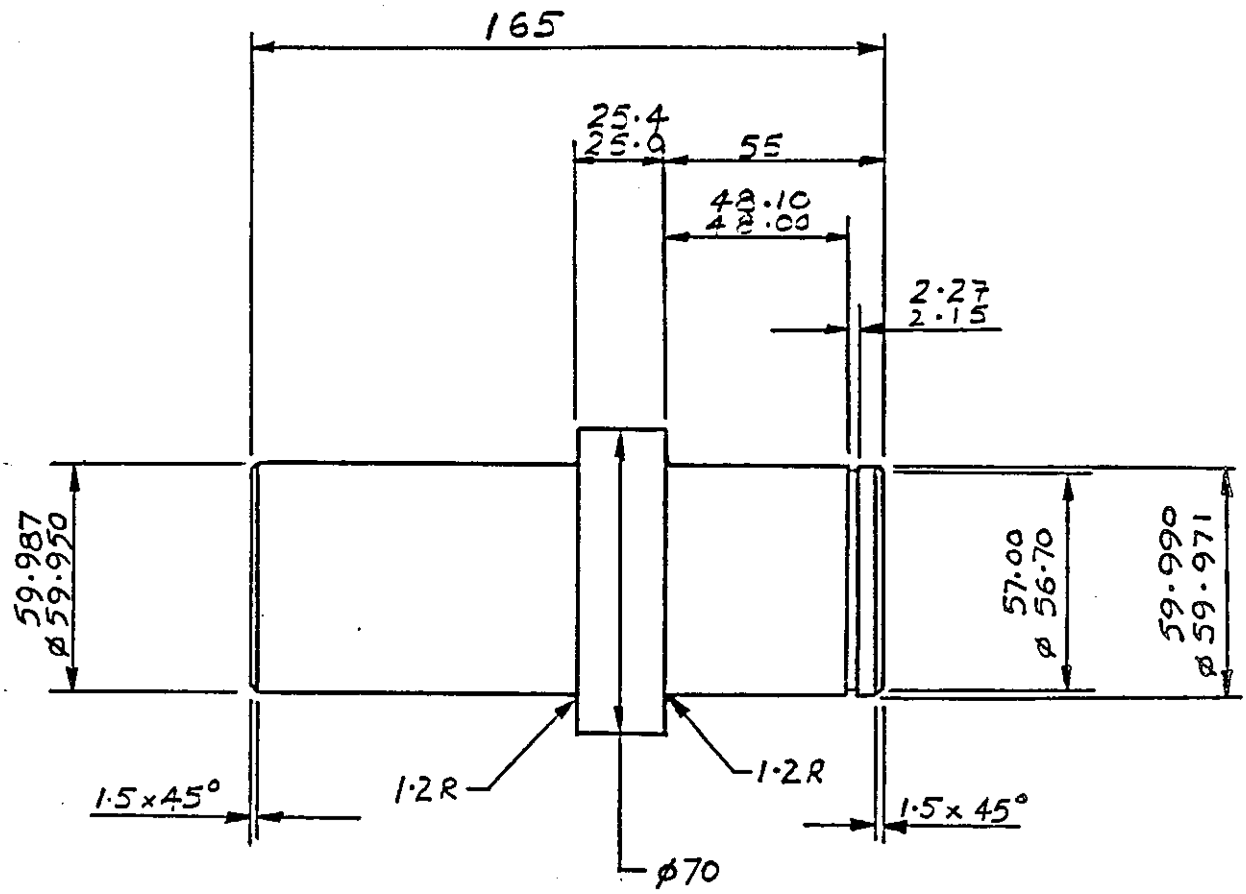
The exact composition of these will vary according to the organization in question. Depending on the size of the organization and the markets it supplies, several functions may be incorporated into one, or individual functions further sub-divided.

For example, work study is essentially concerned with monitoring and improving the efficiency of manual work through activities like method study and work measurement. Clearly in large labour intensive organizations, work study is going to be a major activity, while in smaller highly automated organizations it might not even exist.

In some organizations, several of these tasks may overlap. For instance, in certain circumstances, process planning may be found to overlap NC program generation. In other cases costing and estimating is seen to be integral to process planning. Whatever the particular structure of a manufacturing organization is found to be, in the engineering sector process planning usually has a major influence on the way components are produced. Clearly many other functions are important, but from the engineering rather than the managerial point of view, process planning is dominant.

Once again, there is noticeable variety in the way that process plans are derived and presented by different individuals and organizations. Though for the most part, the process plan will contain information specifying the machine tools, tooling, set-up, technique, machining parameters, materials, job times, costings, operations and sequences necessary for manufacture.

The format adopted for the process plan in many organizations, consists of an extract or reduced copy of the component detail drawing annotated with operation sequences and instructions (see fig 2.13). The level of guidance given in the process plan may vary according to the skill of the workforce or managerial philosophy, for example pursuing a policy of job enrichment that requires operatives to take more responsibility for machining decisions. Therefore, in some cases process planning may be viewed as nothing more than an idealized guide to manufacture whereas in others it will be slavishly adhered to.



PROCESS MASTER

Description

12.5T Runner Pin

Operation 1

Face Centre Turn 59.990/59.971 Leave Grinding Allowance .25

Chamfer Groove 2.27/2.15 Wide X 57.00/56.70 Diameter Part Off to 165 lg.

Operation 2

Hold in collets Turn 59.987/59.950 Leave Grinding Allowance .25 face Chamfer & Centre

Fig 2.13 A typical format for a process plan: Davy Morris

For example in the aerospace industry, the rigorous safety considerations and high technical difficulty generally associated with aircraft components, requires that process plans are explicitly defined and exhaustively followed.

Although in practice a significant amount of process planning is still done manually, the crucial nature of this activity has led to the growth in numbers and use of Computer Aided Process Planning (CAPP) systems. Though CAPP systems have not yet been as successful in industry as other CAD and CAM systems, the view is widely held that the successful implementation of CAPP could form the keystone of an integrated CAD/CAM system. Consequently, a more detailed treatment of the subject is undertaken in the next chapter.

3 CONTEMPORARY APPROACH TO PROCESS PLANNING

Process planning has hitherto depended on individuals of considerable engineering skill and experience for effective implementation. However, even the most considered human decisions are prone to variation over time and between different individuals. Combine all of this with the average time taken to generate a process plan manually and the potential gains from automation are obvious.

The first stage of computerization can be identified in the form of 'variant' process planning, which may be compared with word processing, bringing all the benefits of speed and efficiency associated with it. However, as with the word processor, variant systems are primarily concerned with the efficient storage, retrieval, editing and presentation of existing work. Although this is undoubtedly labour saving, the quality of the process plan is still heavily dependent on the skill and judgement of the individual using the system.

Perhaps it is the complex nature of the logic and knowledge required to produce a process plan that has so far precluded it's true automation. Attempts have been made in the form known as 'generative' process planning which contain many of the key features, though a fully generative system has yet to be produced.

Generative philosophy goes beyond office automation into the realm of automated manufacture and represents the highest level of process planning automation. However there are a number of systems that fall between these two extremes. In order to discriminate between systems at a detailed level it is necessary to adopt a more subtle classification and therefore, it is proposed that four classes of process planning system are identified:-

(1) Variant
(2) Variant-generative
(3) Semi-generative
(4) Generative

3.1 Variant process planning

Variant process planning systems were the vanguard of computerization in the field of process planning. Initially as little more than specialized word processors, they were able to improve upon the effciency with which documentation consisting of text and drawings could be produced by process planning engineers. Whilst no attempt is made to influence or guide the user on the selection of procedures or processes through the variant approach, an indirect benefit of improvements in consistency of presentation and logic is to be expected.

Typically, variant systems rely on the ability to seperate components into classes of similar geometric characteristics. For this purpose Group Technology principles are commonly applied.

Group Technology, as already mentioned in chapter two, is founded on the philosophy that many problems are similar in nature and that by grouping them together, a single solution can be found that saves time and effort. More specifically this can be stated in engineering terms as identifying the existence of families of like natured components. The attributes used in the demarcation of family members will vary according to the application in question, whereupon a coding system is then designed to enable reference to its constituents. In practice many different classification and coding systems are to be found, an example of the technique used in CAPP (29) is the Opitz code (30).

Once order has been established a computer is used to reference, store and manipulate standard process plans, in the form of data files, which can be used as the basis for new plans. In many cases the concept of standard part family components is employed as in CAPP and MIPLAN (32,33), whereby the range of attributes defining each part family is compiled into a single hypothetical model. The model can be used as the basis for a specific part process plan by extracting only the relevant process planning text and data.

Although undoubtedly useful data management tools, variant process planning systems do little to help with the definition of process plans, merely speeding up their composition. Even using standard family process plans, an experienced engineer is still required to define the originals and operate the system. All of which represents a considerable amount of skilled manual effort and it is the dependence on skilled human labour with its attendent inefficiency that the goal of computer integrated manufacture seeks to supplant. In criticising the variant approach, it is important to remember that early systems did not have access to the technology available today.

3.2 Variant-generative process planning

Early attempts to automate process planning have resulted in several systems that are not exclusively variant or generative in nature and can only be described as variant-generative. The discrimination between variant, variant-generative and generative sytems is difficult and subjective, therefore it seems wise to define the reasons for describing a system as being a variant-generative system rather than one of the two common definitions.

Variant and generative represent two extremes, the former a technique based largely on editing the details of existing process plans and the latter a technique for automatically generating a unique process plan from first principles every time. In many cases including a system in one of these two, completely misrepresents it's nature. Clearly the distinction is insufficiently sensitive and therefore a median term is needed which is 'variant-generative'.

One discriminating point is whether the system places more emphasis on the revision or creation of process plans. A variant system would place the majority emphasis on revision, a generative system of course on creation. However, some systems have a creation emphasis but still support a full range of variant capabilties and many second generation variant systems exhibit automatic features which distinguish them from earlier variant systems.

Introducing generative logic to computer process planning systems depends on the ability to program production rules and knowledge of the manufacturing environment, which can be extremely complex. Therefore many generatively influenced systems limit the manufacturing domain to specific processes, in the hope of reducing the problems to a manageable level. Constraints are often applied limiting processes to rotational or non-rotational, and this invariably means turning or milling, similarities to other processes notwithstanding. AUTOCAP (34) is just such a system, specifically aimed at process planning for turned components, whereas ICAPP (35,36,37) is an example of a system limited to components needing milling type operations only. Some systems do adopt a more general approach and an example of the type is GENPLAN (38,39), a self-confessed variant-generative process planner.

Variant-generative systems are still essentially variant in nature mainly using the composite family part concept with group technology classification and coding. However, an ingress of generative philosophy is apparent in the form of specific generative features for calculating supporting machining criteria and/or invoking 'canned' (from NC jargon for commonly repeated cycles of code where parameters only are varied, eg tapped holes) fragments of process plans.

A slightly different approach identified at this level of automation is a technique used in C-PLAN (40), which does not use group technology. Instead, processes, machines and operations are grouped into a hierarchical structure, which the process planner can select via menus designed to control progress through the structure to a logical conclusion. Text held in the system and parameters supplied by the user, are 'type set' to form the finished process plan.

One thing all variant-generative systems have in common is the facility to edit, store and retrieve existing plans. Perhaps more than any other single feature it is this which indicates that a system is variant in origin. Clearly they also rely considerably on the control of a skilled process planner.

3.3 Semi-generative process planning

In order to further discriminate between process planning systems it is necessary to isolate a number of systems which although creation orientated, are deficient because a significant amount of manual effort is still required to produce each new process plan. Commonly the manual tasks are related to initial component description and reflect a low level of integration with CAD, such systems are more accurately described as 'semi-generative'.

Semi-generative systems exhibit many different techniques for user interaction and data input, some of which are very user friendly but still rely on manual input and cannot be considered as truly generative systems.

Input formats range from code based on GT to descriptive languages having much in common with NC part programing languages. MITURN (42) is an example of a semi-generative process planning system using descriptive language for component initialization. AUTOPLAN (43) by contrast employs neither, instead components are defined in terms of co-ordinate geometry.

Semi-generative systems still depend considerably on human intervention in the planning process and it is this feature that distinguishes them from generative systems. Ultimately, no generative system to date can claim to be totally independant of human supervision, but are significantly more so than semi-generative systems. For example, AUTOPLAN does not always produce an optimal manufacturing sequence for the components it analyses, moreover components consisting of external and internal turned features must be defined and analysed as if they were two separate parts. Clearly this would be an impractical and inefficient technique to employ in industry.

3.4 Generative process planning

A generative process planning system is commonly defined as one which synthesizes a plan from first principles every time. In the ideal case a generative system would be a complete substitute for manual process planning. In so far as no system yet demonstrates the ability to completely duplicate the role of manual process planning, no truly generative system exists to date. This conclusion is supported by Steudel (59). However, the pursuit of generative process planning has produced many systems exhibiting a high degree of automation which is central to the approach.

If the range of components or manufacturing processes to be planned for is limited in some way then the objectives of generative process planning become more attainable. Thus several systems might be said to have achieved this goal within a specialized domain.

The first requirement of any generative process planning system is a comprehensive component description and it is at this stage that present generative systems need human assistance. Beyond this point several systems demonstrate a high degree of automation that is consistent with the generative approach.

Systems such as CAPSY (44) are reconciled to the human link and are user friendly as if by way of compensation, making good use of computer graphics and dialogue for interactive communication. By contrast a 'no compromise' philosophy is used in TIPPS (50,51) which is designed around a boundary model of the kind used in many 3D CAD systems, bringing it as close as any to full CAD integration.

With respect to the confidence of Chang and Wysk (52) it is interesting to note a significant aspect in which TIPPS fails to demonstrate true potential for an automated link with CAD, namely the recognition of machined surfaces from a 3D CAD model. In TIPPS machined surfaces have to be identified manually, by selecting from menus of different shaped surfaces then using a cross-hair cursor to indicate connected faces.

Recent work by Choi et al (60), on the automatic recognition of machined surfaces from a 3D solid model, concludes that a complete automation of process planning may not be possible because of problems related to machined surface recognition.

SHAPES (45,46) has a similar approach to TIPPS, implying if not demonstrating a potential CAD link, it is arguably one of the more specialized systems. SHAPES contains a full specification of the capacities and attributes of a small manufacturing cell encoded into the system, enabling precise process planning. An ability to test design features against manufacturing capacity makes it possible within SHAPES to provide feedback to the designer if they are exceeded.

SHAPES also exemplifies the concept of the 'tool orientated approach' described by Lewis et al (49), which is formulated on the premise that component features imply tooling and that tools can be ordered into tool hierachies rather like the 'canned cycles' found in NC programing. A simple example of this principle is the tapped hole, where a sequence of centre drilling, tap drilling, taper tapping, second tapping and plug tapping is usual. Hence one feature can indicate a fixed cycle of many operations. In this simple example the designer will already be aware of standard sizes for tapped holes and plan his design accordingly. The extension of this principle to less obvious, or preferred, standards for production gives feedback likely to improve the awareness and performance of design for manufacture.

An interesting difference in contemporary generative philosophy is highlighted in PROPLAN (53), which departs from standard practice in so far as it treats process planning as the inverse of machining, working backwards from the finished component to it's stock material. This approach can simplify the logic because operations are taken in a strict sequence. Conventional planning often requires several operations to be considered in parallel to achieve the desired finished condition. In PROPLAN this activity is referred to as 'devolution', whereas other systems using the approach, TIPPS for example, use a different name.

PROPLAN is a system capable of general application, although in reality this means it is by no means suitable for all engineering components. Although integration with CAD is claimed, and in this case CAD stands for computer aided draughting, this is only achieved through use of the 'house' CAD system (PADDS) which is unique in its approach. PADDS allows information about the engineering features of a component to be incorporated into the drawing model at the draughting stage, which proves valuable to the subsequent activity of process planning.

LOCAM (54,55,56) is also a system intended for general application, based around a modular approach that is designed to introduce an organization gradually to generative process planning. The arguments for a staged introduction of this kind are compelling, because in reality any company initiating or evolving computer aided process planning, will learn from the experience of building up the company logic necessary to emulate manual process planning. Until each stage of increased automation is reached and understood an organization is unlikely to be able to specify the next step accurately.

This appears to imply that by starting from fundamental relationships, a higher level of abstraction from the detail of process planning is reached with each stage of automation, until the manufacturing concepts involved can be inferred directly from design. Possible drawbacks are the amount of time and work needed to implement such a structure, the lack of standardization between organizations, as well as potential inflexibility to new technology, because of the constant necessity to update complex system logic.

Similar drawbacks apply to XPS-1/2 (57,58) which through its origination on the CAMI process planning programme has sought to be universal in application. Inevitably a considerable amount of additional work is necessary to implement systems of this kind because they have to be tailored to suit an individual organization's manufacturing requirements.

3.4.1 Expert systems

Recent interest in the potential for the application of expert system techniques to computer aided process planning has prompted systems like EXCAP (47,48) and STOPP (65). Such systems depart from the more conventional algorithmic and database approaches, prefering instead to represent process planning logic as a set of 'rules of thumb'. The rule based approach used in EXCAP has a further refinement, the ability to attribute a 'fuzzy value' to each rule. In basic terms the fuzzy value is an estimate of the confidence with which a rule can be applied and seeks to emulate the nature of human knowledge, which is rarely absolute. Although in practice many rules can be applied with a high degree of confidence in algorithmic form, there are occasions when knowledge is imprecise and algorithms are inappropriate.

Knowledge based or 'expert systems' are a result of research into artificial intelligence (66,67,68,69,70), designed to tackle the problems of incomplete and imprecise kowledge in fields of human expertise. The result of successful implementation can be an improved performance over the human expert because of the innate precision of computation. A very good example of this occurred in the field of medical science, where an expert system dealing with diagnosis called MYCIN (71), could consistently outperform it's human counterpart. It was reasoned that although a doctor had the same knowledge, he could not recollect, sift and process it as efficiently or as accurately as a computer.

To take the MYCIN example a little further, the domain of its supremacy lay in one particular aspect of clinical diagnosis and represented a number of years of development work to attain it. The problems of developing a system to replace the knowledge needed by the average GP are orders of magnitude higher. The moral of this experience may be that it is feasible to encode efficient knowledge bases for a specialized domain but not for general application at the present level of technology and knowledge.

Research into knowledge based process planning systems is at an early stage, examples of such systems (47,48,61) are specialized in application and lack the well rounded appearance of their conventional counterparts. The expert system approach to process planning does not seem likely to produce a general process planning system within the near future.

There are some very real obstacles to implementing a totally generative approach, arising from the magnitude and complexity of the task. A risk exists that the necessary knowledge will continue to evade researchers for some time to come. For example, almost since the first computer was built, computer scientists have been trying to develop natural language translation systems, although always seemingly possible, success eludes researchers to this day.

Having already compared the problems facing practical exploitation of knowledge based process planning systems to those already encountered by conventional ones, it is interesting to note that Spur et al (44), express a belief that planning work which is based on creative considerations can hardly be automated and regard the computer as an aid to, rather than a replacement for, manual process planning. In a similar manner Lewis et al (49), originators of the tool orientated approach, used incidently as the basis of EXCAP, admit that it cannot duplicate the expertise of an experienced process planner.

By contrast there are authors of generative systems, Chang and Wsyk (52) for example, who are confident that a universal fully generative system is ultimately feasible, although even they concede that a greater understanding of the nature of manufacture and its implications for process planning is needed before this goal can be attained. In detailing these prerequisites, it becomes apparent that a complete solution is some time away.

3.5 Variant process planning systems

Before proceeding any further it is important to describe some examples, starting with variant systems. These include CAPP (29,30,31) and MIPLAN (32,33), both of which use Group Technology based classification systems as the key to identifying similar parts.

The CAM-I Automated Process Planning (CAPP) System was developed by MCAUTO under contract to CAM-I (31). It is an interactive software system to assist process planners in the generation of manufacturing planning documents. The use of a Group Technology based classification and coding system such as OPITZ (30), is intrinsic to the approach.

CAPP takes as a starting point, standard process plans for each of the part families identified during classification. These take the form of a set of machining instructions that could be used to create a process plan for any part in the family.

In order to draft a process plan for any given component, its code is initially used to identify the relevant part family by referring to part family matrices, which tabulate the attributes of the constituents of each part family. An example is shown in table 3.1. From the standard family process plan, a basic operation sequence is defined which can be further refined to produce a part dependant process plan. This is then stored on the computer and can be accessed for review or editing, should the process planner wish to use it as the starting point for planning another part without regressing to the standard family process plan.

MIPLAN, currently marketed by the Computervision Corporation, was originally developed in 1976 by the Organization for Industrial Research (OIR) as part of a modular CAD/CAM system. It facilitates the creation of process plans by using text and drawings raised interactively by the planning engineer. If parts are coded into families by the MICLASS system then variant planning is possible; three modes of use can thus be identified:-

(a) A process plan may be created from scratch using standard process descriptions held in text files which can be assembled and edited as is appropriate for the part in question.

(b) An incomplete process plan can be retrieved from the computer file and finished.

Part family name	Part family number
Plain bushings and spacers	Ø1ØØ
Step bushings and spacers	Ø2ØØ
Precision bushings and spacers	Ø3ØØ
Pulleys	Ø4ØØ
Hydraulic fittings 1	Ø5ØØ
Hydraulic fittings 2	Ø6ØØ
Angle fittings	Ø7ØØ
Forks and studs	Ø8ØØ
Levers and arms	Ø9ØØ

Table 3.1 Part families used in CAPP

(c) An entire process plan can be retrieved by indicating the whole/partial classification code or according to criteria appearing in the process plan 'header'.

A more recent development by OIR called Multicap extends the number of modes to four:-

(d) The process planner can begin with a simple sketch and code the part using 'Multiclass' to enable retrieval of similar parts.

Multicap extends MIPLAN into variant-generative form, without this enhancement, MIPLAN is very similar to CAPP and both are typical of the variant approach.

3.6 Variant-generative process planning systems

Examples of variant-generative systems are AUTOCAP (34) for rotational components, ICAPP (35,36,37) for non-rotational, GENPLAN (38,39), C-PLAN (40) and CAPES (41), which are all systems of more general application.

3.6.1 Rotational systems

AUTOCAP is a development of earlier work at UMIST on a computer aided process planning system for turned components. It uses an interactive approach to converse with the process planner, in order to obtain the necessary information for generating a process plan. Amongst other things, the user has to specify the order in which component features are to be machined, so the term 'generating' must be interpreted rather loosely.

Some help is given in the aspect of operation sequencing and work holding considerations by reference to a predefined composite master part (see fig 3.1) in the manner of a variant system. Once this has been determined, each feature of the component is taken in the required cutting sequence and described to the system by a code and related parameters, such as tolerance, surface finish, starting and finishing dimensions. Whereupon company dependant databases are interrogated to generate the remaining machining criteria. The ability to store, retrieve and revise existing process plans is also supported by the system.

3.6.2 Non-rotational systems

ICAPP is an interactive process planning system for prismatic parts, originating from UMIST. The system is feature orientated, with the capability to process eight different geometric features based on common machining operations, as

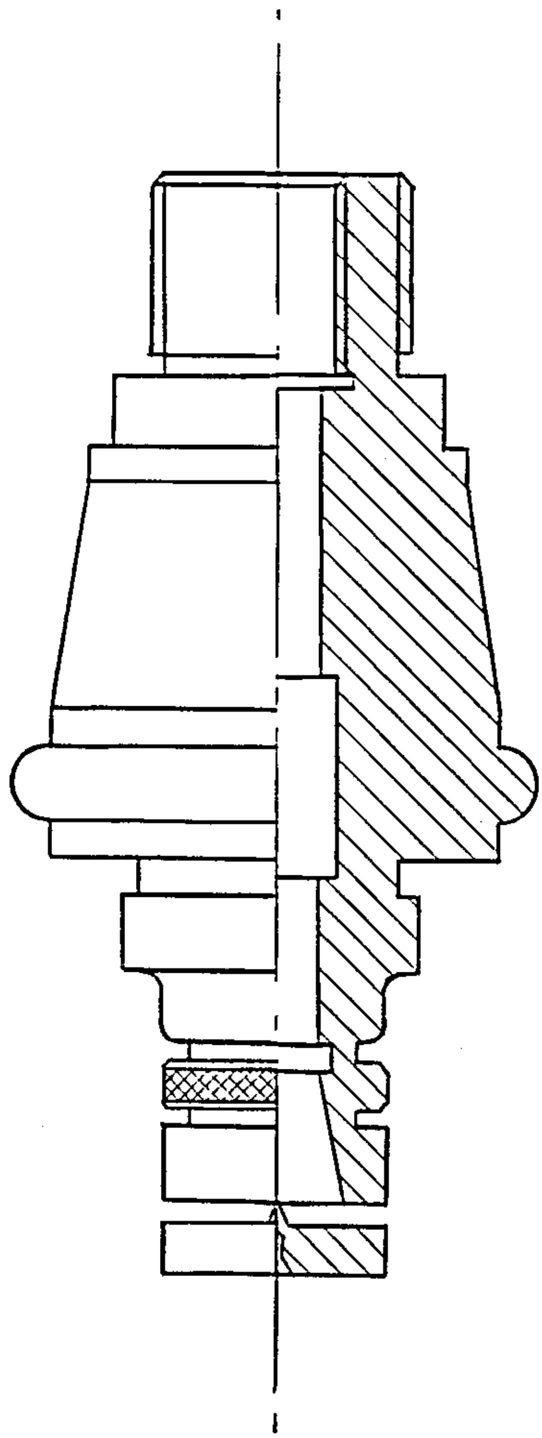


Fig 3.1 AUTAP composite master part

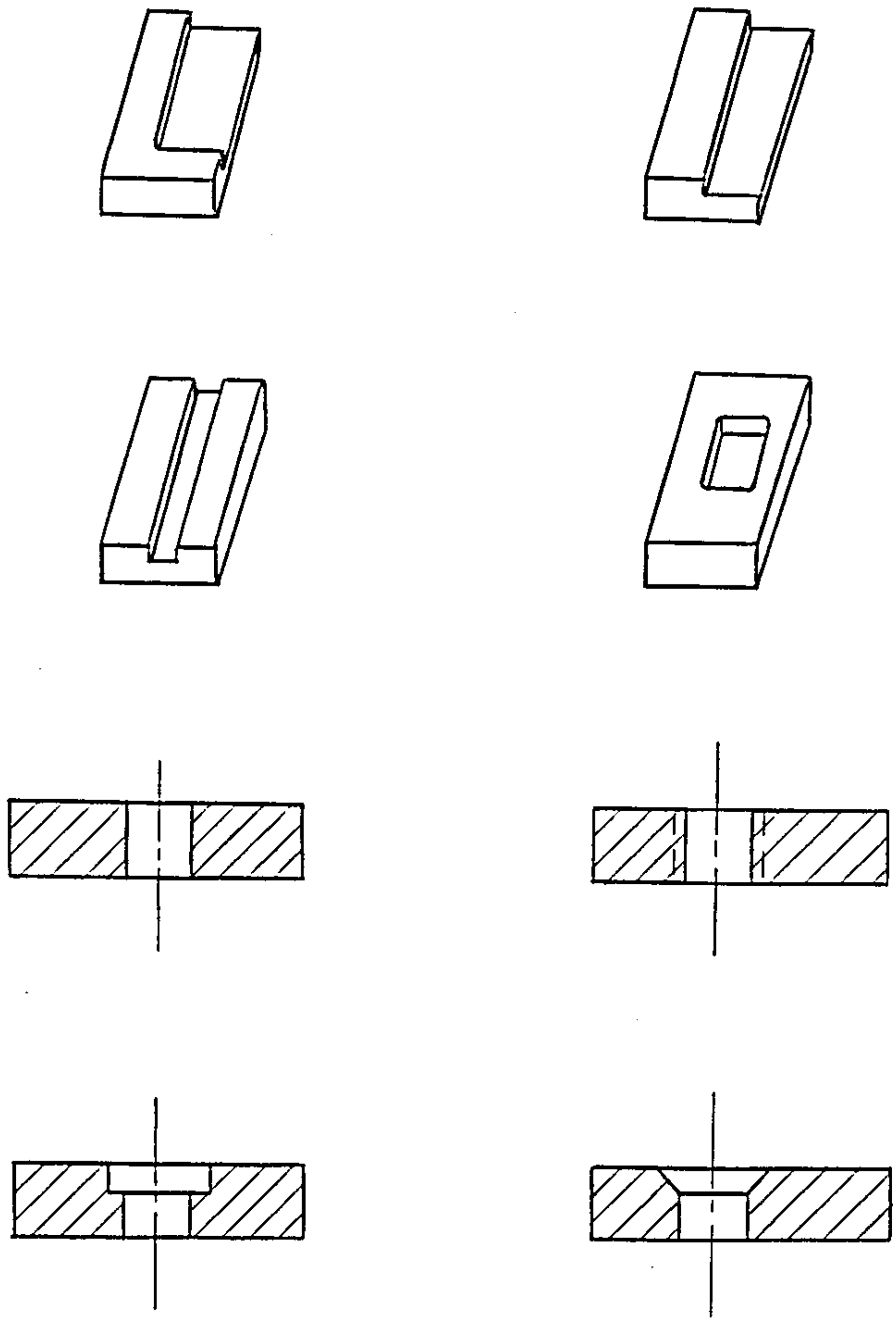


Fig 3.2 ICAPPS's eight geometric features based on common machining operations

in fig 3.2. When deriving a process plan, suitable machining operations are selected by the system according to feature type dimensions and tolerances input by the process planner.

System logic is a combination of variant planning, via the part family concept (composite part), with supporting generative logic. A full description of the component must be entered as there is no conspicuous CAD link, although an extension has been made to enable production of NC tape. The supporting geometric and dimensional information must also be entered interactively.

3.6.3 General systems

GENPLAN, was developed by the Lockheed-Georgia Company, although short for 'generative process planner' it is admitted to be really a variant-generative system because a trained process planner is still required to operate it.

To raise a process plan, the user has to identify the relevant code in a specially developed group technology based classification system. The code which encompasses part geometry, size, and manufacturing processes is entered on the computer, from which basic planning decisions are generated. The remaining work is completed by the process planner, which is claimed to be 'minor fill-in' only. Facilities are also provided for editing and supplementing existing process plans in the variant style.

C-PLAN produced by CADCentre, Cambridge and CAPES by INBUCON, are similar systems, representing a slightly different approach to process planning automation. Unlike AUTOCAP and GENPLAN, they do not use group technology classification and coding systems, although additional software modules provide an interface to the 'house' CAD system.

Using C-PLAN the engineer creates process plans by a simple dialogue with the program using menus and 'forms' (for parameter values to be entered). The menus group processes, machines and operations into a hierarchical structure, thus by stepping through the menus and indicating the desired options a process plan is composited (literally, as the text from the menus is 'type set' along with it's dependent parameters for printing the process plan). An example of this menu driven approach is shown in fig 3.3.

A standard company database has to be appended to the core system detailing manufacturing capacity, though the abilty to specify standard calculations and planning logic elevates it to a level of automation sufficient to qualify as variant-generative. Variant features are also evident in the facilities provided for edit ing, storing and retrieving existing plans.

1 Material preparation
2 Turning
3 Milling
4 Drilling
5 Presswork
6 PCB assembly
7 Processes
8 Inspection

1 Capstan lathe
2 CNC lathe
3 Turret lathe

1 Load to collect
2 Feed to stop
3 Rough and finish turn __mm dia X __mm long
4 Turn thread dia __mm X __mm long
5 Drill __mm dia X __mm deep
6 Tap __mm dia X __mm deep
7 Rough and finish bore __mm X __mm deep
8 Face __mm dia
9 Part off __mm dia
10 Chamfer edge
11 Pack to avoid damage
12 Unload
13 Drill __ hole X __ depth
14 Rough turn __ X __ length

Fig 3.3 Example of C-PLAN menus

3.7 Semi-generative process planning systems

Typical semi-generative process planning systems are MITURN (42) and AUTOPLAN (43), which are both designed to deal with rotational components.

MITURN was developed at Metaainstituut TNO, Netherlands for use in process planning for NC lathes. It uses a part programing language, similar to the kind used in NC programing, to communicate with the process planner. The component to be planned must first be described to the system using this language which consists of listing it's 'elements' consecutively starting from the tailstock end, where an element is an internal or external component feature like a diameter, groove, taper and so forth, with it's attendant parameters (see fig 3.4).

Once the part description has been input the system determines the cutting sequence, tooling, machining parameters, machining times and produces the NC tape.

AUTOPLAN, a joint effort between the Imperial College of Science and Technology and Glamorgan Polytechnic, is outwardly similar to MITURN though instead of a part programing language, component 'sections' (synonymous with MITURN's elements) are defined by co-ordinates on the two-dimensional parting-off plane, comprising the axis of

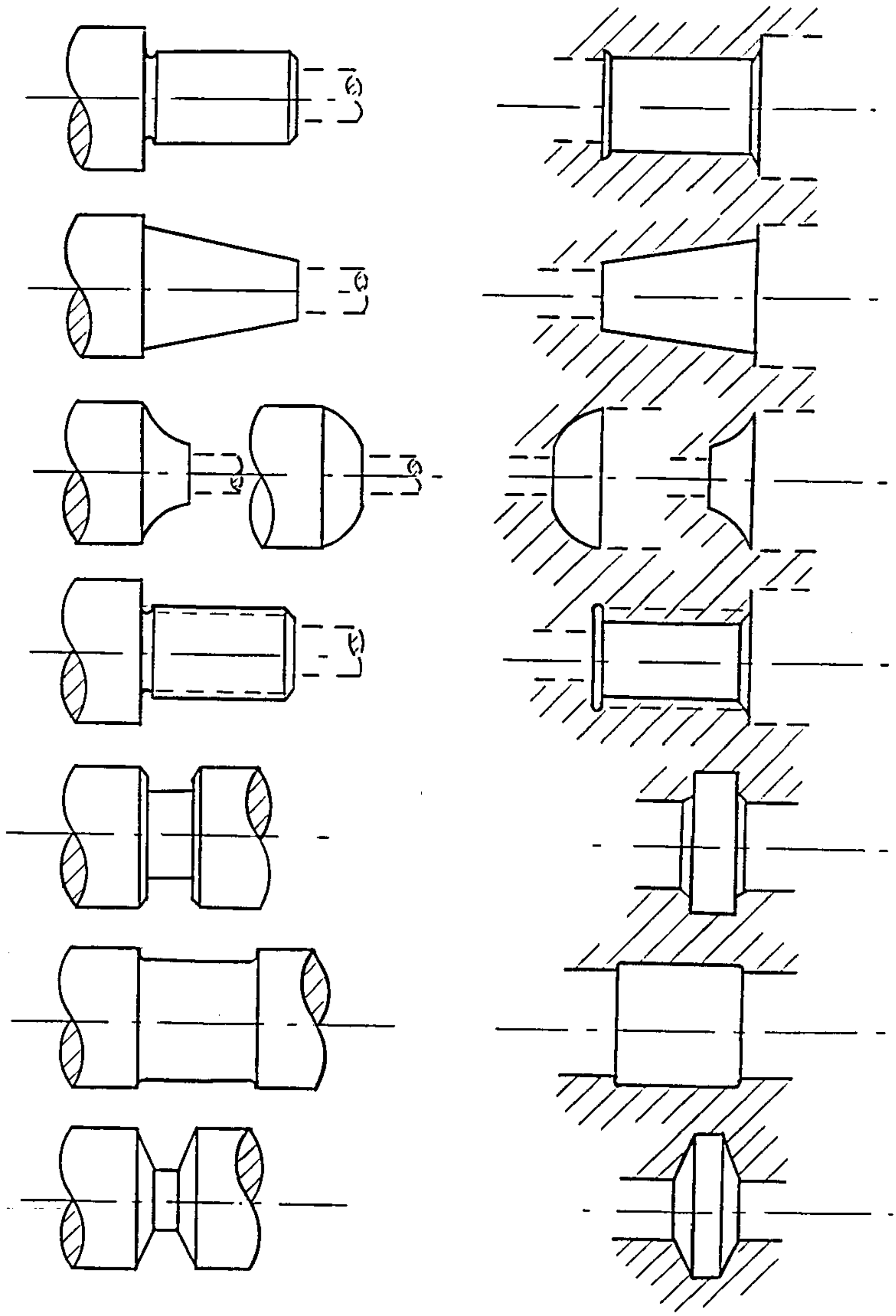


Fig 3.4 MITURN's internal and external elements

symmetry and the radial axis (see table 3.1 and fig 3.5). Internal and external features are defined seperately, thus components having internal features are divided into two seperate parts.

The process planning logic is programed in FORTRAN in algorithmic form and generates textual output. This does not necessarily represent the optimum manufacturing sequence, particularly in the case of components having internal and external geometry because of the necessity to treat such components as two seperate parts. A degree of post-processing is still essential.

3.8 Generative process planning systems

As before, examples of systems described here fall into three categories: rotational, non-rotational and general according to the manufacturing processes covered by them.

3.8.1 Rotational systems

CAPSY (44), SHAPES (45,46) and EXCAP (47,48), are systems that fall into this category.

CAPSY was developed at Berlin University, described as a dialogue system for process planning it is currently limited

External data

POINT	XO(1)	YO(1)	SECTION	TOLER	FINISH	TPIO(1)
1	000.0000	000.0000	1	11	50	–
2	000.0000	000.2500	2	13	51	–
3	000.2500	000.2500	3	13	51	–
4	000.5500	000.5500	4	11	52	–
5	001.0000	000.5500	5	13	50	–
6	001.0000	000.4000	6	13	51	–
7	001.5000	000.4000	7	11	51	–
8	001.5000	000.3000	8	13	50	–
9	002.0000	000.3000	9	13	50	–
10	002.0000	000.0000				

Internal data

POINT OR SECTION	XI(1)	YI(1)	TOLER	FINISH	TPII(1)	TAPANG(1)
1	001.0000	000.0000	13	50	–	150
2	001.1250	000.1250	13	50	–	–
3	001.2500	000.1250	13	50	–	–
4	001.2500	100.1250	13	50	16	–
5	002.0000	100.1250	13	50	–	–
6	002.0000	000.1250	13	50	–	–
7	002.0000	000.0000	13	50	–	–

Table 3.2 Example of input data required by AUTOPLAN

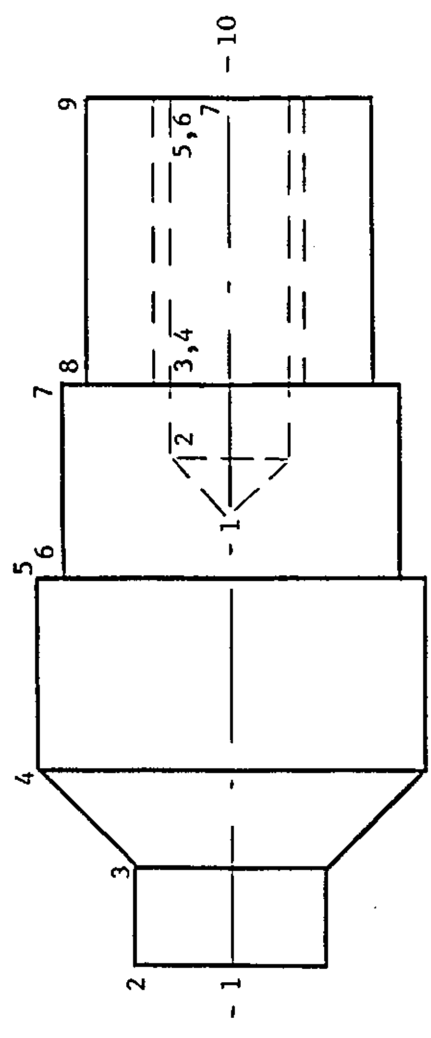


Fig 3.5 Example of input reference points required by AUTOPLAN

to turning and drilling processes. Considerable emphasis is placed on the use of computer graphics for interactive communication with the system, which makes it easier for the user to monitor planning decisions, enter supporting data and detect problems.

The system is intended to use descriptions of parts and blanks prepared in a 3-D modelling system like COMPAC (26,27), which gives it a relatively high level of CAD integration consistent with the generative approach.

The complete system is divided into processors for various machining processes organized in a hierarchical structure. The ability to define company specific data and logic in any combination with the processor modules makes the system appropriate for wide application.

The system is resigned to the fact that manual intervention is still required for the definition of permissible machining areas, examination of work holding considerations and limited data input. In this respect, CAPSY is like most other generative systems.

A joint effort between UMIST and Heriot-Watt University has resulted in a system called SHAPES, an attempt at producing an automated process planning system for a small manufacturing cell of three CNC turning centres. A full specification of the capacities and attributes of the cell is encoded to enable a precise comparison between component features and production capacity.

A feature orientated input is used to describe the component to be planned in terms of its elemental shapes (see fig 3.6), internal or external geometry, with supporting dimensional and material information. Although not yet present, the capacity to interface with CAD systems is implied.

The system processes the component description, analysing its features in respect of the encoded machine capacities to ascertain whether or not it can be manufactured and if so by which combination of machines, tooling and work holding devices. Feedback is given to the designer if component details are specified that exceed the programmed capacity of the cell. Process planning logic follows the tool oriented approach.

EXCAP is the result of more recent work at UMIST, which incorporates expert system technology into a computer aided process planning system. In its present form it is designed to plan rotational work only, although it is claimed to be equally capable of application to other processes because of the flexible nature of such 'rule based' knowledge systems.

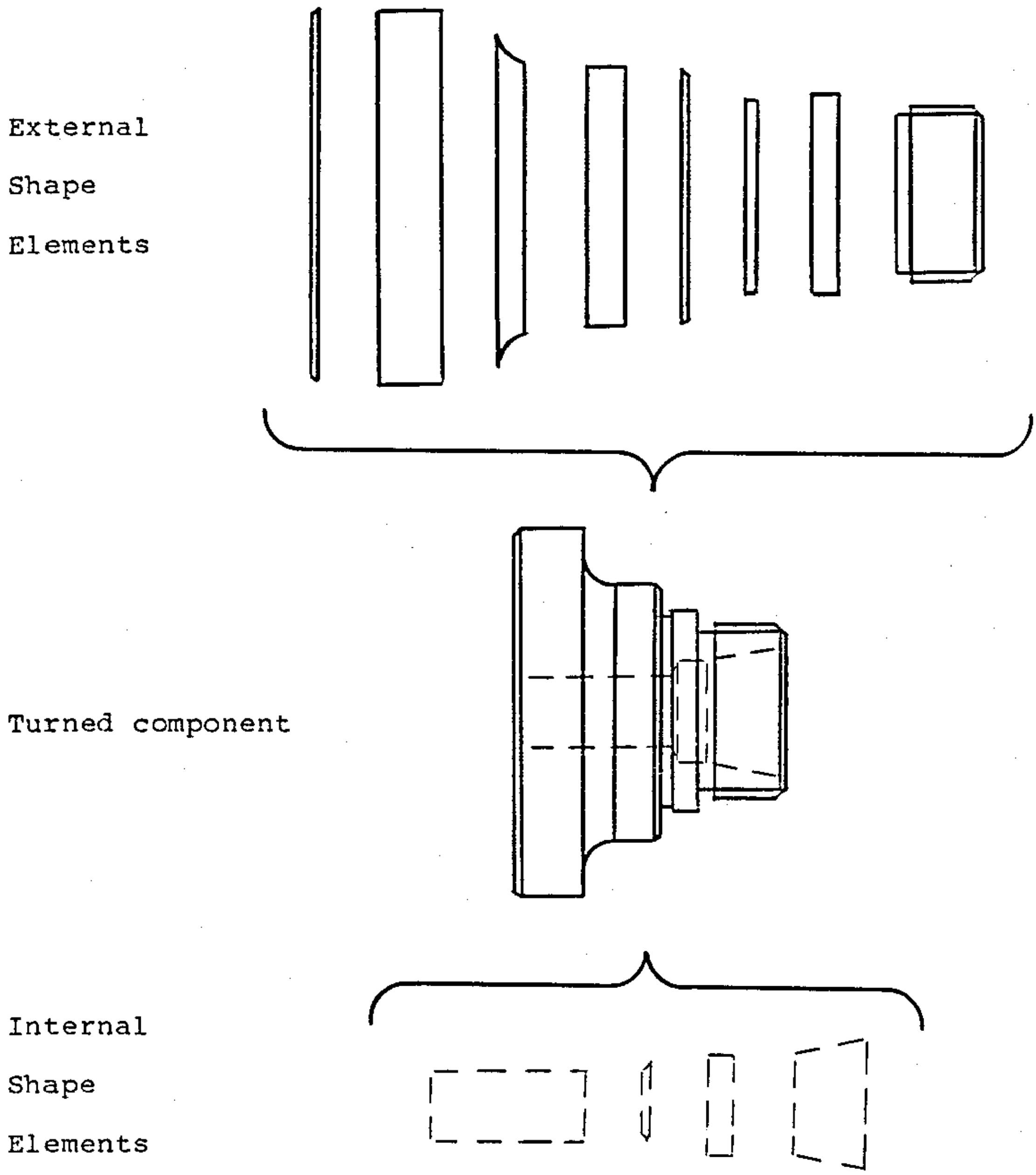


Fig 3.6 'Shapes' shape elements

The rule base represents an encoded form of the process planners expertise in a manner that is distinctly different from the more conventional algorithmic and database approaches. It has the ability to emulate the imprecision of much human knowledge, representing knowledge in relative rather than absolute terms, thus:-

IF	(to a certain extent)	(condition)
THEN	(to some degree)	(action)

This technique is more appropriate, it is argued, to process planning because of the lack of success to date in identifying accurate algorithmic logic for this activity.

The rules used in EXCAP perform two distinct functions. They relate machining operations to component features, describing the effect that the operation has on the workpiece, as well as evaluating the most suitable operation among those applicable to the task. The planning process itself is also split into two phases. A macro phase for determining operation sequencing, then a micro phase in which machining parameters are calculated for each macro operation.

EXCAP departs from standard practice in so far as it treats process planning as the inverse of machining, working backwards from the finished component to it's stock material. An approach which can simplify the logic, but is clearly an abstraction of reality.

The imprecision of process planning is also reflected by an ability to generate different permutations of the process plan from which an optimum must be identified. Current limitations of the system preclude serious application, for which considerable extension of the rule base would be necessary and direct communication with CAD systems.

3.8.2 Non-rotational systems

TIPPS (50,51) is the third generation system of an evolutionary chain including APPAS (52) and CADCAM (52), all developed jointly at Purdue and Pennsylvania state universities.

TIPPS is particularly notable for the degree to which it can be interfaced with CAD systems, as it uses a boundary model of the kind used by many 3-D CAD systems for an internal component representation. Interactive graphical techniques are used by the process planner to label the surfaces of the component to be machined according to type. Process and sequence information is then generated, taking the supporting dimensional information directly from the boundary model.

Process planning knowledge is encoded using a special purpose language called Process Knowledge Information (PKI) which appears to be a purpose written FORTH vocabulary. This translates as essentially a rule based approach similar to that used in EXCAP, but without the capacity to attribute 'fuzzy value' (an estimate of the confidence with which rules can be applied). Similarity also occurs in the way TIPPS uses inverse logic to deduce the process plan by 'adding' material to the finished component rather than 'removing' it from the stock material (see fig 3.7).

In practice components that can be processed by the system are limited to box-shaped components with holes and machined surfaces approached from the top surface of the part. Interrogation of the CAD styled internal representation is not always sufficient to determine surface relationships and on these occasions further user input is required.

3.8.3 General systems

PROPLAN (53), LOCAM (54,55,56) and XPS-1 (57,58), are examples of generative systems having wider application.

A system has been developed at the Production Engineering Research Association (PERA), for turned, drilled and most milled components, a range which is claimed to include 50 percent of components made in the engineering and allied industries. The system is called PROPLAN.

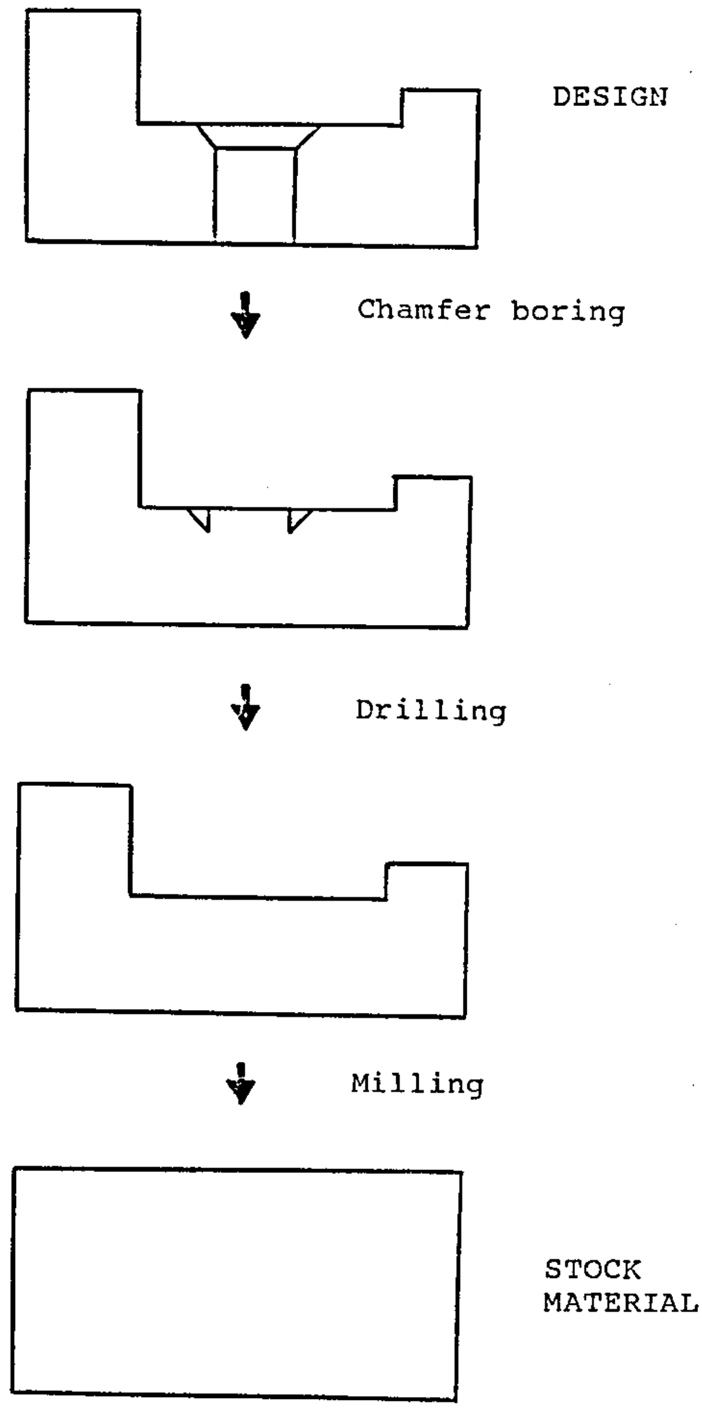


Fig 3.7 Example of 'devolution' or inverse planning

PROPLAN is designed as a module to form part of a comprehensive suite of programs leading towards a fully integrated computer aided manufacturing system. In nature it can be described (in the terms used here) as a conventional system, using a database and algorithmic approach. If used with PERA's own computer aided drawing system PADDS, it is claimed that a fully integrated design to process planning link is possible. However, this ability stems from the unconventional nature of PADDS, which enables manufacturing information to be included in the component drawing that can be read by PROPLAN.

The manufacturing information which must be supplied to PROPLAN in support of the 2-D drawing consists of engineering attributes such as tolerance, surface finish and hardness. Special features like threads, gear teeth, knurling, keyways and tapped holes, to name a few, are also treated as attributes. Each geometric feature of a component described in PADDS has associated with it a variable number of attributes, therefore if PROPLAN is used with any other computer aided drawing system, attributes must be determined during an intermediate processing operation.

In addition to the component and raw material models the system needs data about the manufacturing facilities available to the organization in question. This data must be established and coded into three files on machine tooling, cutting tools and work-holding equipment.

PROPLAN uses an inverse machining or 'devolution' approach similar to EXCAP and TIPPS, whereby a components attributes are first eliminated/reduced until primary methods of manufacture remain. Devolution is completed by adding material to the component on a geometric basis until the raw material condition is reached. The process plan which is finally generated may vary from the devolution sequence if subsequent interrogation of the organization's databases indicates that an alternative procedure could reduce the number of seperate operations. An extract from the PROPLAN process plan for the component shown in fig 3.8 can be seen in fig 3.9.

LOCAM, developed by Logan Associates, is marketed by Prime Computers as a suite of software modules that can be integrated and extended into a 'fully generative process planning system'. The modular approach is designed to encourage users to grow with the system as the emphasis is on the company to build in information and logic describing the manufacturing system employed. Thus further modules encourage a staged increase in the level of process planning automation.

At the recommended entry level (stage 2), the system is variant-generative in nature, driven by a combination of

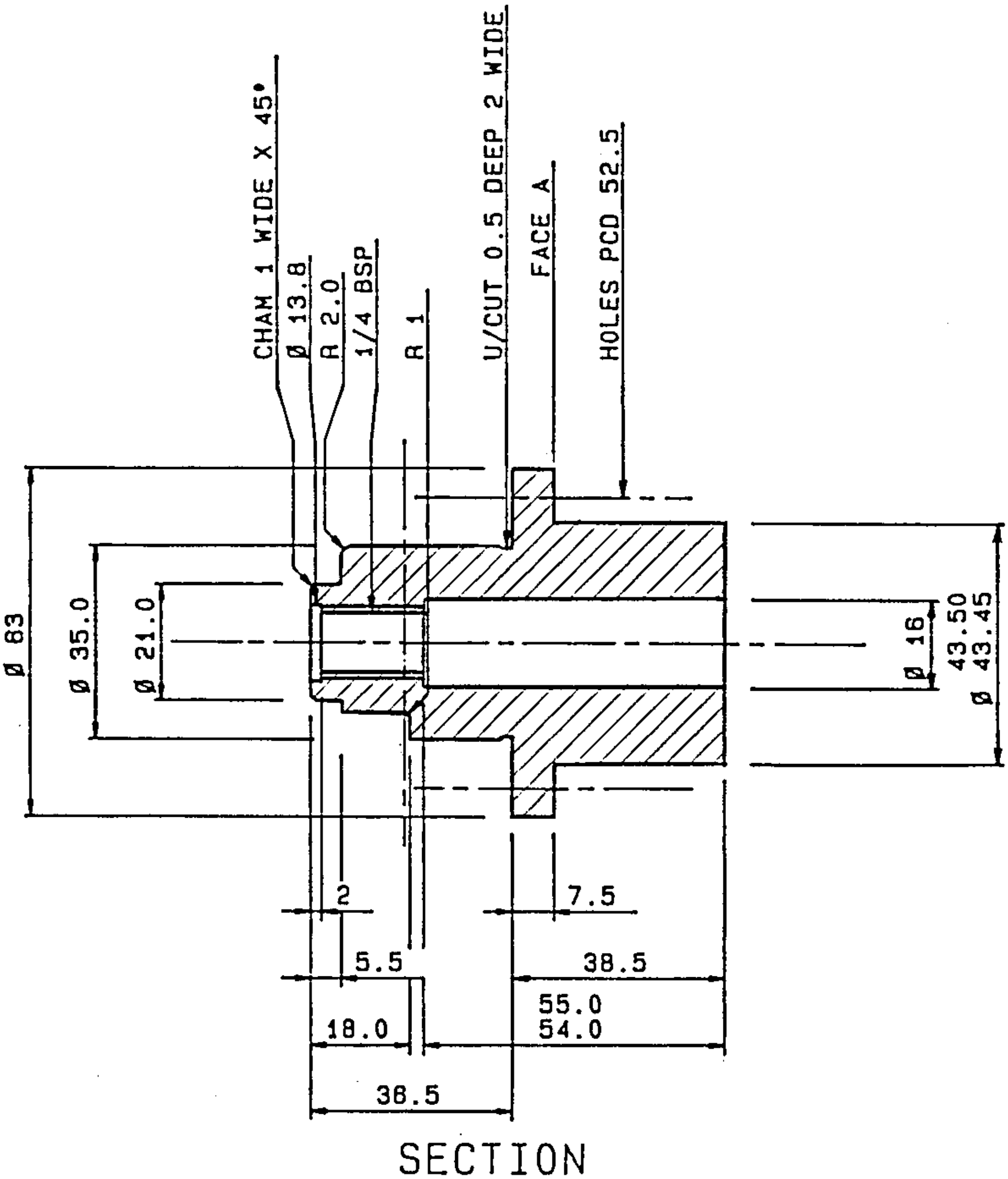


Fig 3.8 Component used in PROPLAN process planning example (see Fig 3.9)

Date: WED, 22 DEC 1982	All dimensions in MM unless otherwise stated	
Part no: 64133	Part name: VALVE BODY	
No. parts per blank: 1	Blank size: 65 dia X 77 long	

OPERATION CONTENT	TOOL	M/C GROUP
OP NO : 10 : TURNING Load to 3 jaw chuck (hard jaws) Face end to clean Centre drill BS328 type A pilot dia 2 Drill 11.4 dia thro. Turn 35 dia from 65 dia X 36.5 long Turn face from 35 dia to 65 dia Turn 63 dia from 65 dia X 7.5 long Turn 21 dia from 65 dia X 5.5 long Turn face from 21 dia to 35 dia Turn u/cut 2.00 wide X 0.50 deep to blend with face on 35 dia Turn 2 male corner rad to blend on 35 dia Turn 45 deg chamfer X 1 wide on 21 dia Turn bore from 11.4 dia to 13.8 dia X 2 long Turn face in bore from 13.8 dia to 11.4 dia Tap 1/4" BSP thread X 18.5 deep		
OP NO : 20 : TURNING Load to 3 jaw chuck (soft jaws) Face end to 75.00 length Turn 45 dia from 65 dia X 31 long Turn face from 45 dia to 65 dia Fine turn 44.00 dia X 31.00 long Fine turn 43.50 dia X 31.00 long Turn bore from 11.4 dia to 16 dia X 54.5 long Turn face in bore from 16 dia to 11.4 dia		

Fig 3.9 An extract from a PROPLAN process plan

question/answer dialogue and predetermined keywords with associated parameters encoded into a company standard manufacturing database. Stage 3 uses the database to identify questions/keywords that can be grouped together and described as a single feature or attribute. This concept is further extended in stage 4 where features are grouped in a similar manner to compose a code for component production. The interfacing of a classification and coding module to a CAD system, such as 'Medusa', incorporates into the design process the capacity to generate component codes. At this level, the system becomes semi-generative.

Stage 5, which is the level of generative process planning, is seen as the natural conclusion of fully implementing stage 4 using complex part coding linked to CAD. Clearly this would require considerable time and effort to achieve, moreover the degree of success would vary considerably according to the diversity and complexity of the particular manufacturing enviroment concerned.

The experimental planning system version one (XPS-1) evolved out of the CAM-I process planning programme, instigated in 1974 to investigate and develop methods for automating and standardizing process planning. Earlier work by CAM-I which resulted in the variant system CAPP, has already been described, XPS-1 (57) followed and subsequently an improved version XPS-2 (58).

XPS-1/2 is perhaps best described as the compiler of a computer language for process planning. Planning logic is captured by the system through the use of a specially developed CAM-I process planning language (CAPPL). The XPS-1/2 compiler then processes these English-like statements into executable code. This activity is refered to as decision modelling.

Supplementary to the decision models, a data dictionary and relational database are used to incorporate information on the manufacturing enviroment, for example machine tool capacities and tooling data. A classification and coding system is also incorporated into XPS-1/2 to relate the decision logic to actual components, using the part family concept.

In operation three levels of use are identified as shown in fig 3.10. The provision of menus and help information enables a user friendly interface to be built into the system, but there is no evidence of an ability to interface with a CAD system.

3.9 Other process planning systems

Of course there are many other process planning systems that have not been mentioned in this literature survey. This is

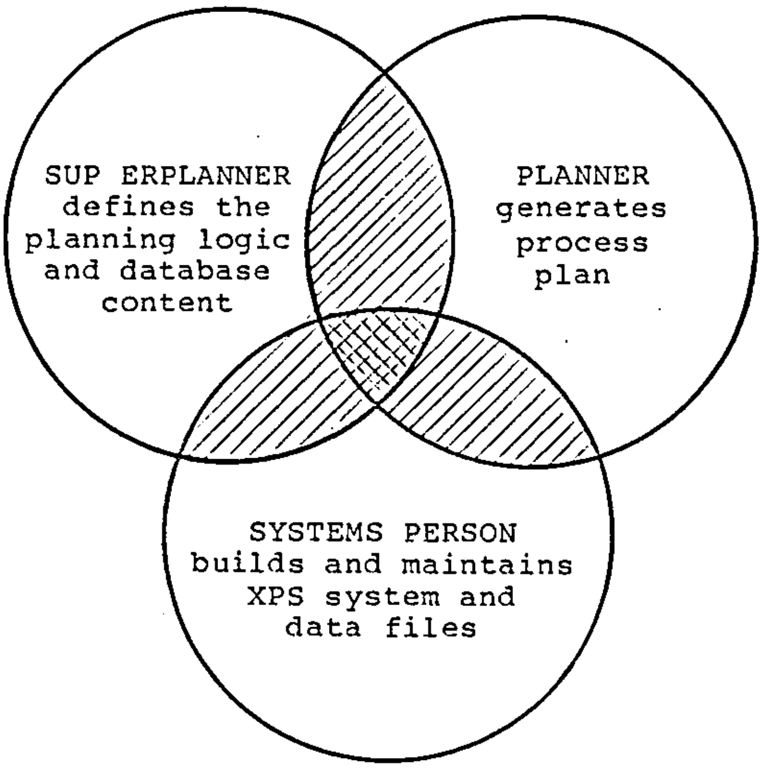


Fig 3.10 Three levels of using XPS 1/2

not to say that they are of no interest, just that the examples given are probably better known and adequately illustrate the issues concerning contemporary computer aided process planning approach.

Otherwise, acknowledgement must be given to many more systems, like AUTAP (62), a generative process planning system for rotational and sheet metal parts, developed at the University of Aachen in Germany. AUTOPROS (63), which was developed at the University of Trondheim in co-operation with Norwegian industry, represents an ongoing project for automating process planning. In China a variant-generative system called TOJICAP (64), uses group technology and standard plans of master parts to generate new process plans. There are many more.

4 A NEW APPROACH TO INTEGRATING DESIGN AND PROCESS PLANNING USING FAMILIAR ENGINEERING LANGUAGE

So far, the issues concerning contemporary approach to computer aided process planning have been presented and constructively criticized. Consideration has also been given to present philosophy on the CAD to CAM link and the views and proposals of many researchers in the field on the direction that future research should take. It is apparent to many that there is a need for some fresh thinking on this subject.

Much emphasis is being placed at present on the future use of expert system technology for solving the problem of an automated CAD to CAM link. However, this must be seen as a long term objective as progress is likely to be slow for the same reasons that conventionally structured computer aided process planning systems have encountered. Mathematically, there are still gaps in our knowledge about aspects of representing and interpreting fully featured 3D computer models. Possibly more significantly, our understanding of the manufacturing environment in general is still very limited and thus the capacity to encode it, in whatever form, must be similarly limited.

Presented here is an alternative view for making a successful CAD to CAM link that is not primarily concerned with the techniques for encoding knowledge. Instead the view is taken that a fundamental change in the present approach to design may be beneficial to wider manufacturing objectives.

4.1 Practising design for manufacture principles

Although the principle of design for manufacture is becoming widely accepted as a cost effective means of improving overall maufacturing efficiency, the extent to which it is actually practised is less so. The reason for this could be that the traditional structure of manufacturing organizations does not intrinsically encourage it. Take for example the typical structure:-

Design --> Process Planning --> Machining

In many organizations, all three of these activities are completely separated managerially and, more often than not, physically. It is not uncommon to find poor communication and even an attitude of hostility between departments. This separated approach has continued to some extent into the process of computerization, with systems developed independently and aimed at specific sectors of the manufacturing structure. Not surprisingly, similar communication problems exist between computer systems, for it is hard to achieve complete electronic integration without first understanding the problems of full manual integration.

In order to practise design for manufacture it is essential that there is a mutual understanding and co-operation between design and process planning disciplines. This would best be achieved if the two were interwoven, manually at first and ultimately electronically. This is not to suggest that other disciplines in the manufacturing structure should not be so integrated, just that the focus of this thesis is on the specific relationship between design and process planning.

To design for manufacture effectively, it is proposed that the designer must in part moot an outline process plan (see fig 4.1) for the production of components (73). Although a designer is unlikely to be concerned with peripheral details, he must consider primary operation sequences and production processes. The generation of this information, presents considerable computational problems for contemporary computer aided process planning systems. In fact it is one area in which all generative systems at present are unsuccessful.

The logical conclusion of this argument is that if it is beneficial to consider an outline process plan at the design stage, it is extremely illogical and inefficient to discard the information only to subsequently synthesize it.

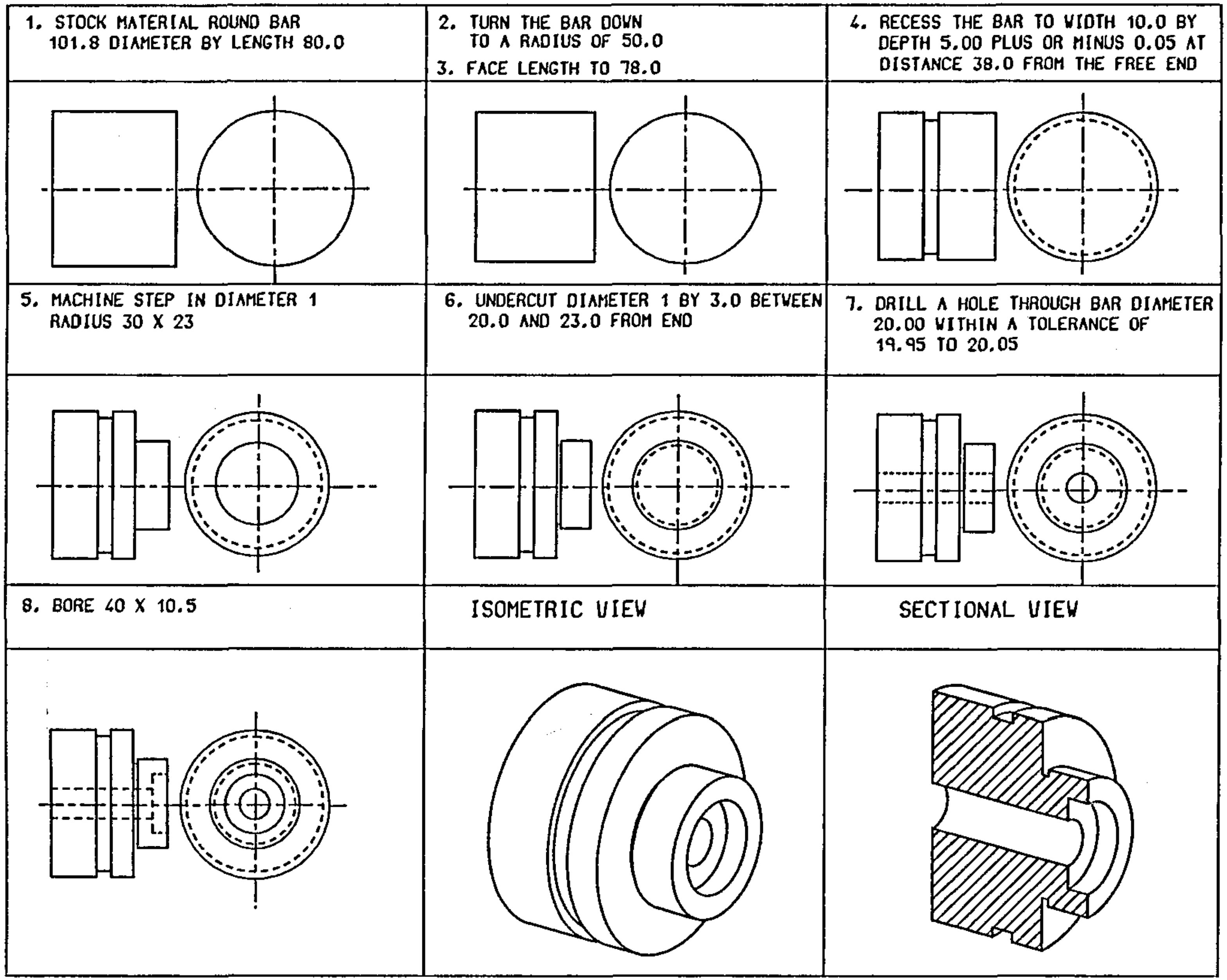


Fig 4.1 Example of an 'outline process plan'

Particularly, when the synthesis of the information requires enormous computational effort.

This view is controversial because many would instantly argue that either the designer is not competent to plan for production, or that such an arrangement is highly undesirable because of the increased workload and responsibility placed on the designer. These objections can be countered.

The sensible application of computer technology can significantly reduce the time expended on design calculations and presentation, giving the designer more time to consider manufacturing problems. The use of computerized databases can increase the accuracy and speed with which design and manufacturing information can be accessed, making it easier to design for manufacture. Also, the knowledge required by the designer to define an outline process plan is arguably fundamental to his job, if design for manufacture is to be practised.

The absorption of process planning into design need not necessarily be a one way process and is certainly not a mandate for making process planners redundant. Instead, in the same way that designers would be encouraged to take some responsibility for process planning, it should be possible to integrate the skills of process planners into the design office, in terms of labour as well as knowledge. No precise formula is ventured for the manner in which this might be accomplished, because in the event it would depend on the organizational structure in question. The important thing is to adopt a flexible approach, for it is pointless to pursue flexibility in manufacture if it starts and stops only on the shop floor.

The implications for computer aided process planning are that it would not be necessary to struggle with the problems of feature recognition, primary process selection and operation sequencing, because such information could be extracted and recorded during the design phase. Take for example the as yet unsolved computational problems of automatically recognising machined surfaces from a solid model identified by Choi, Barash and Anderson (60). It is inconceivable that such machined surfaces are not blatantly obvious to a designer.

A computer aided process planning system would still be needed to fill in specific information to satisfy manufacturing needs, such as speeds, feeds and tooling. Much of this could be determined using suitable algorithms, although the possible use of knowledge based techniques is not ruled out. Focussing attention on the aspects of computer aided process planning that generate the details nec essary to support an outline process plan, so reduces the computational task that it would be realistic to produce a fully design integrated system within a comparatively short time.

In reality the process planning system would not be a seperate independent entity, but fully integrated with the design modelling system. The basis for which must almost certainly be a 3D geometric modeller, as the only feasible way of modelling a component to the level of detail required. Nevertheless there are some obstacles to overcome if this approach is to succeed.

4.2 Interaction with a 3D geometric modelling system

Reference has already been made, in chapter two, to the role that a 3D geometric modelling system might play in a fully computer integrated manufacturing system. It can be concluded that 3D geometric modelling is the only technique with the potential of representing a complete engineering description of a component in computer terms.

At their present stage of development, there are a number of obstacles to implementing a practical system centred around the existing 3D geometric modelling systems which are not exclusively concerned with mathematical theory.

Adopting a geometric modeller as the heart of an integrated system may allow the definition of component geometry in a method analogous to manufacture and provide a mathematical description for future reference. However, it does not allow manufacturing information to be included in the model. Moreover, typical command languages of present modelling systems are alien to designers and engineers, as they can often consist of boolean operations and complex coordinate geometry. Although some modellers use a menu driven approach that is more user friendly, the technique is better suited to the input of geometric information than it is to the input of manufacturing details.

A new kind of communication interface is called for to ease the description of components and obtain salient manufacturing information. For these reasons and to ensure wide user familiarity, such an interface might best consist of familiar engineering language.

4.3 A command language for engineering

A major consideration in the unification of design and manufacturing information is that while representation of design information is ostensibly graphical, manufacturing information is predominantly textual. Ultimately they may share a common binary format but as long as there is a manual link in the manufacturing chain computer systems will need to communicate this information in understandable language.

For input, output, or programming purposes, the format is usually a specialized subset of common language. Just such a language already exists in the engineering industry, which apart from certain variations in 'dialect' is universally understood. It's standards are not as precisely controlled as the engineering drawing but it would seem to be a suitable starting point from which to define an engineering command language. Obviously a degree of standardization would be desirable but an intrinsic flexibility would also be of considerable benefit in terms of wide and immediate user familiarity. This flexibility is an essential feature of a language serving different levels of a manufacturing organization.

A familiar engineering language might have several applications in an integrated design and process planning system:-

(a) As an input language to supplement graphical techniques.

(b) As a controlled internal enviroment to represent manufacturing information.

(c) As the means of generating output suitable to form a process plan.

(d) To facilitate interrogation of manufacturing information by non-specialist users.

Since manufacturing information often consists of detailed textual descriptions, the use of a textual input language is likely to be of benefit. It is not obvious how the techniques typically used in CAD systems, for example menus, joysticks/mice and digitizers, can simplify the input of detailed manufacturing information. Therefore, the use of a familiar engineering language to supplement these techniques would seem necessary.

If manufacturing information is to be incorporated into a complete electronic description for a component along with geometric data, called the 'product model' by Sata et al (72), then a method of encoding it must be found. The use of vetted input, as provided by a familiar engineering command language, would ensure consistent quality and quantity of information for use in 'downstream' applications.

The familiar engineering language, might be ~~be~~ used not only to monitor input, but also to generate output. For instance, data from other sources might be expressed in familiar engineering language. Thus output in a form appropriate for use in a process plan could be produced regardless of the method in which it was entered.

As many types of users are likely to seek manufacturing information from the product model, for example general management, the use of familiar engineering language is likely to be understood by more people.

Whatever the exact use to which a language of this kind is ultimately put, the content and definition of it is potentially complex and ambiguous. Therefore, considerable importance has been attached to the practical testing of this ideal.

4.4 Language development and structure

As a starting point, three existing formats for command languages were identified and considered on their merits:-

4.4.1 Rigid structure

This is the typical format used for most computer programing languages, for example FORTRAN, where a precise syntax and semantic form is defined which the user has to learn. There is little licence for choice of diction, ordering of information or customization, although the structured nature of such languages ensures unambiguity.

The careful choice of command names and associated parameters makes it is possible to create suitably meaningful 'jargon'. In the case of BOXER (23), a solid modelling system which uses this kind of command language, the jargon is of a mathematical nature. A simple solution might then be to supplant BOXER's mathematical jargon with an appropriate engineering jargon, for example:-

```
object name  length  radius
     |            |     |
     C1 <- CYL (100, 25)
             |
  object type (cylinder)
```

might become:-

```
COMPONENT = ROUNDBAR (LENGTH=100, DIAMETER=50)
```

Obviously this would be more easily understood by someone with an engineering bias rather than a mathematical one. However, further problems arise when the tasks become more complicated and additional information pertaining to, say process planning, is incorporated. One possible solution might be to allow comments as in FORTRAN or remarks as in BASIC, which are ignored by the compiler/interpreter so that information can be included that is not related to the command structure, for example:-

```
COMPONENT = ROUNDBAR (LENGTH=100, DIAMETER=50)
(use bright drawn mild steel, turn, case harden and grind)
```

Clearly the diversity of such commentary is beyond the scope of a rigid command structure. Also, the relegation of such information to mere commentary raises serious doubts about the likelihood of it being either of the necessary quality or quantity. Experience suggests that information of no direct relevance to the task in question (in this case generating the model) becomes highly suspect and of little value. A major disadvantage considering the application in question.

4.4.2 Keyword searching

Keyword driven programs are a step towards natural language programing and if well implemented can be impressive in their apparent comprehension of complex commands. For example, applications include ELIZA (74), a psychological program using keywords for conversational pattern recognition, and less seriously, the so called 'adventure' computer games.

By definition the command structure used in keyword searching only discriminates between certain words and ignores everthing else. This can present serious problems when the subtleties of syntax and nuances of semantics are ignored because the potential exists for complete misinterpretation.

Even after thorough examination of the subtle aspects of a keyword command structure there still exists the problem of 'non-specific' words (words that of themselves do not singularly and unambiguously define an object, activity or quantity etc.) that can take on key importance, for example:-

TURN THE BAR DOWN BY 5.Ø MM

TURN THE BAR DOWN TO 5.Ø MM

Here keywords might appear to be 'turn','bar', and 'down', combined with the dimension, but the words 'to' and 'by' though non-specific have crucial influence on the meaning of the sentence. It is hard to see how these factors can be tackled successfully without using a high level of syntactic and semantic analysis, which is contrary to the aims of keyword searching.

4.4.3 Natural language

Natural language programming is the aspect of computer science concerned with the interpretation and communication of human language, for example English, by a computer. It is a truly massive subject and the most successful attempts to date do little more than highlight the complexity of the task. Current approaches tend to concentrate on the parsing of a sentence according to one or more rules of syntax and/or semantics. Charniak and McDermont (75), conclude that programming problems notwithstanding, few linguists even agree about the form that a grammar for English should take.

Assuming that these problems are surmountable, further complications arise with ungrammatical (wrong form) and agrammatical (no form) constructions. Unfortunately typical familiar engineering language often strays into this category and so appears to preclude a formal natural language approach.

As none of the foregoing promised a realistic solution in their original states, a hybrid format has been selected for the engineering language.

4.5 Hybrid Format

The resulting hybrid format can best be described as 'familiar language' as distinct from natural language with its foundations in linguistic science.

If the absence of sufficient knowledge to enable clear definition of rules for interpreting imprecise grammatical constructions, strongly precludes a natural language approach. Attention to the subtle influences of syntax and semantics also precludes the use of keyword searching, which suggests that a rigidly structured form is called for.

The problem is how to define a format which is both flexible for the user, but rigid and therefore unambiguous to the computer. One possible route is to exhaustively map the language for all possible constructions and encode it. The flexibility will then depend entirely on the quality of the map. As an ideal this is acceptable but a number of problems exist:-

(a) The range of the map is virtually infinite in so far as the scope for variety is concerned.

(b) The ability to map constructions in an unambigious manner is increasingly strained as the complexity of the map grows (constructions overlap).

(c) The combinatorial nature of this approach threatens to overtax computation.

If theoretically these points are discouraging, practically there are modifying factors:-

(a) The map of most commonly used constructions is finite for a specific application with a given target (i.e. satisfy most of the users most of the time).

(b) A degree of standardization of terminology is desirable (e.g. to meet company standards).

(c) By breaking the map into smaller segments, like subroutines in programing, combinatorial problems are kept to manageable levels.

Having overcome these problems, the resulting language is rigidly structured but flexible because it is extensive. Keywords may be identified, but never without complete respect for their context. Where keywords are non-specific understanding occurs by virtue of the culminated meaning of the whole sentence. Admittedly the method owes little to conventional natural language programing, although it does aspire to similar goals, albeit within a confined domain.

4.6 The use of a command processor to develop familiar engineering language

As an aid to the development and implementation of a familiar engineering command language, a software tool known as a command processor is used. Developed and widely used at the Computer Aided Design Centre (CADCENTRE), Cambridge, it is known as GILT (76), which is short for Graph Input Language Translator.

A command processor is designed to process user input into a form which is easily used by applications software. As an independent piece of software itself, it can be integrated into any suitable system, thus forming a consistent user friendly interface for a wide range of software products.

The CADCENTRE command processor is intended to allow users to enter pseudo-English commands, that are both easier to understand and remember. The benefits of using the command processor can be summarized as follows:-

(a) The command processor simplifies the construction of an English-like command syntax for the application program.

(b) It monitors command input, checking validity and if necessary rejecting the input before the application program is aware of it. Thus unreasonable commands are filtered out.

(c) The resulting semantics can be changed readily, often without any need to alter the application program. In this manner, various 'dialects' of familiar engineering language can be accommodated.

Additional benefits acrue from its use that are essential to a system suitable for diverse application. Such features include the provision of a querying facility that enables the user to determine the full range of responses available to him at any time during command input. Help information can also be incorporated into the command language for access by the user, as well as abbreviations and synonyms for user customization.

Although obviously intended as a 'front end' to a system, there is considerable scope for generating highly descriptive output suitable for process planning applications.

In order to represent the syntax and semantics of familiar engineering language in a form suitable to be encoded by GILT, it is helpful to adopt a graphical approach and manually map out the language first (77). For this purpose, a technique has been developed for representing complex language graphs. The approach used, while appropriate to GILT, is also of sufficiently general form to be applied by any suitable technique.

For the purpose of describing the graphical representation of command language and in deference to its application using GILT in this instance, CADCENTRE terminology is now used. The description of the technique is both lengthy and complex, though it is necessary if a full appreciation of the familiar engineering language graphs presented in the next chapter is to be gained.

4.7 A graphical technique for representing familiar engineering language

In GILT a command consists of a sequence of characters terminated by a carriage return (typing enter/return key). In simple terms this can be likened to a sentence, where typing the enter key is equivalent to typing a full stop. For simplicity, no other punctuation is used and characters are limited to alphanumerics only. The format of a familiar engineering command is one of exclusively alphabetic combinations (words) and numeric combinations (signed or unsigned numbers with or without a decimal point) seperated by spaces. Each command represents a seperate manufacturing operation including all the relevant details.

For a command to be acceptable to the system, it must be one of the commands allowed in the language map, matching exactly the syntax and semantics therein. Refering to figure 4.2, in

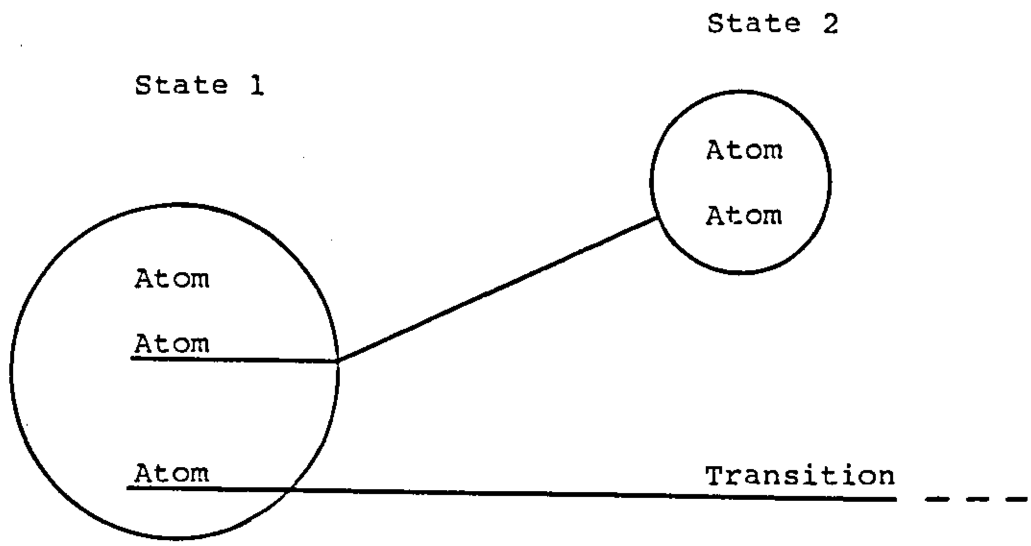


Fig 4.2 Anatomy of a subgraph

graphical form, the simplest possible language map consists of a single GRAPH consisting of ATOM's, STATE's and TRANSITION's.

Atom is the term used to describe any member of the set of words and numbers allowed in the language map. The set of all atoms is therefore the total vocabulary of the language map, rather like a dictionary is the set of allowable English words. States consist of one or more atoms and represent a range of alternatives at that point in the graph. From any particular atom there is a link to the next appropriate state in the graph and this is called a transition.

Graphically, atoms are shown in uppercase text, states as encircled atoms and transitions as lines where the direction of movement is always from left to right. Enlarged arrow heads indicate the entry and exit points of a graph.

The complete language map need not be contained in one graph alone but may take the form of several graphs, chained or nested accordingly, in the manner that subroutines or procedures are employed in conventional computer programming. However this introduces conventions in addition to those described so far.

The name of a graph is printed in lowercase italics and once defined is itself an atom, although rather a special one because it is comprised of other atoms. Graph names may appear anywhere that other atoms are allowed and may themselves label graphs containing other graph names. When a graph name is encountered in tracing a path, the path is diverted to the graph of that name. Progress in the original graph may not continue until a complete path through the named graph has been traced successfully.

It remains to introduce three more special atoms, the first of these is called the NULL TRANSITION. Which, if encountered, diverts the path immediately to another state and is the only time when a state may be exited without matching an atom. Note that this does not mean that any atom is allowed at this point, merely that the atom is carried forward and must be matched in the subsequent state. Graphically, the null transition is represented by a hyphen.

The atom VAL is used to capture a number when a number features in a command to be entered by the user. If val is encountered in tracing a path through a graph, it should be considered as calling an internal graph, predefined within GILT, whose purpose it is to interpret characters at that point in the command as a numeric value. Obviously, if the sequence of characters does not constitute a recognisable number according to the definition mentioned earlier, then the input cannot be accepted. Val, although represented graphically in italics like external graph names, is an internal graph within GILT and therefore the reader will not find it documented further.

The last of the special atoms is NL, signifying new line, which can be regarded in exactly the same way as val except that it is the purpose of the nl graph to match with a carriage return. It is used exclusively, in the application of GILT to familiar engineering language, to terminate commands.

Finally, an explanation of the manner in which the contents of a state should be interpreted seems appropriate. The list of atoms contained in a state must be read in order from top to bottom, for it is in this order that atoms are compared with user input to determine whether a match can be made. Order becomes important in states where null transitions occur because atoms below a null transition would never be processed. Also for clarity, where several atoms require the same transition to another state only one transition is shown. This transition is shown for the lowest atom in the list of atoms concerned and is underlined to accentuate it.

Once completed the graphs are rewritten in a form called Graph Input Language, which is translated by GILT into data that can be used by the command processor.

4.8 An example of GILT applied to tolerancing

The explanation given so far can be further clarified by recourse to a simple example, which is taken from the familiar engineering language map for turning, described in full in chapter five.

The following example deals with the case of tolerancing, prevalent at all stages of manufacturing where dimensions are specified. Take for instance the need, when it arises, to specify a tolerance in relation to a nominal diameter. The text defining a diameter might take any one of the following forms (Note that the word diameter appears here in uppercase, whereas numbers are represented by lowercase names):-

...DIAMETER dimension...

...DIAMETER dimension tolerance...

...dimension DIAMETER...

...dimension DIAMETER tolerance...

...dimension tolerance DIAMETER...

Clearly this text would be embedded into a more complex command and perhaps even repeated several times within the same command, for example, refering to fig 4.3 the turning operation might be described as follows:-

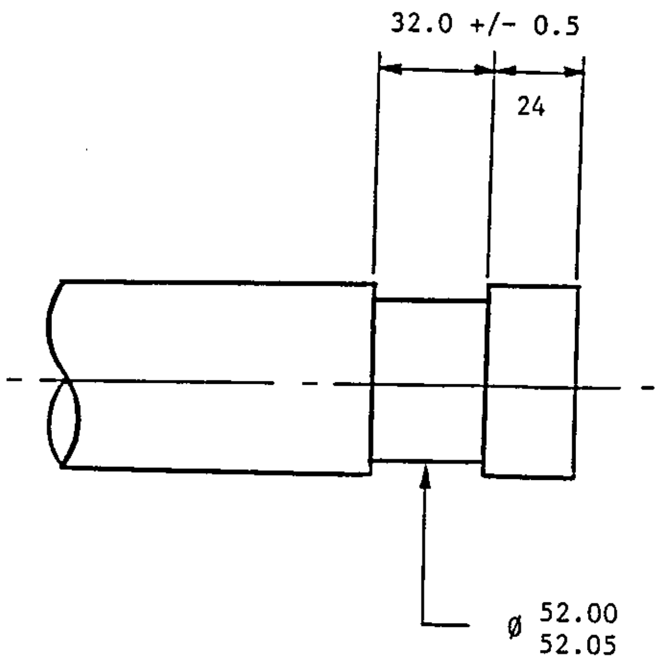


Fig 4.3 Turning a recess in a bar

Turn a recess in the bar with dimensions of width 32.0 plus or minus 0.5 and diameter 52.00 to 52.05 at a distance of 24 from the free end.

Here tolerances are specified on the diameter of the recess and its width, although a different format is used in each case, whereas no tolerance is indicated on the positioning of the recess (probably implying a default to a company standard tolerance for machining processes). Similar commands are constantly repeated throughout the map of familiar engineering language.

Considering that dimensions are captured by val, which is already defined as a seperate graph within GILT because it is so often used, it is also logical to create a seperate graph for tolerancing as shown in fig 4.4.

Once the graphical representation of the tolerance graph has been completed, it must then be converted into graph input language before it can be processed by GILT, this takes the format shown in fig 4.5. If the graph for tolerancing is expanded to show all of the possible command permutations, the result is shown in fig 4.6, anyone of which is equally acceptable to the system.

At this point it is necessary to explain a little more about graph input language. During the process of input validation

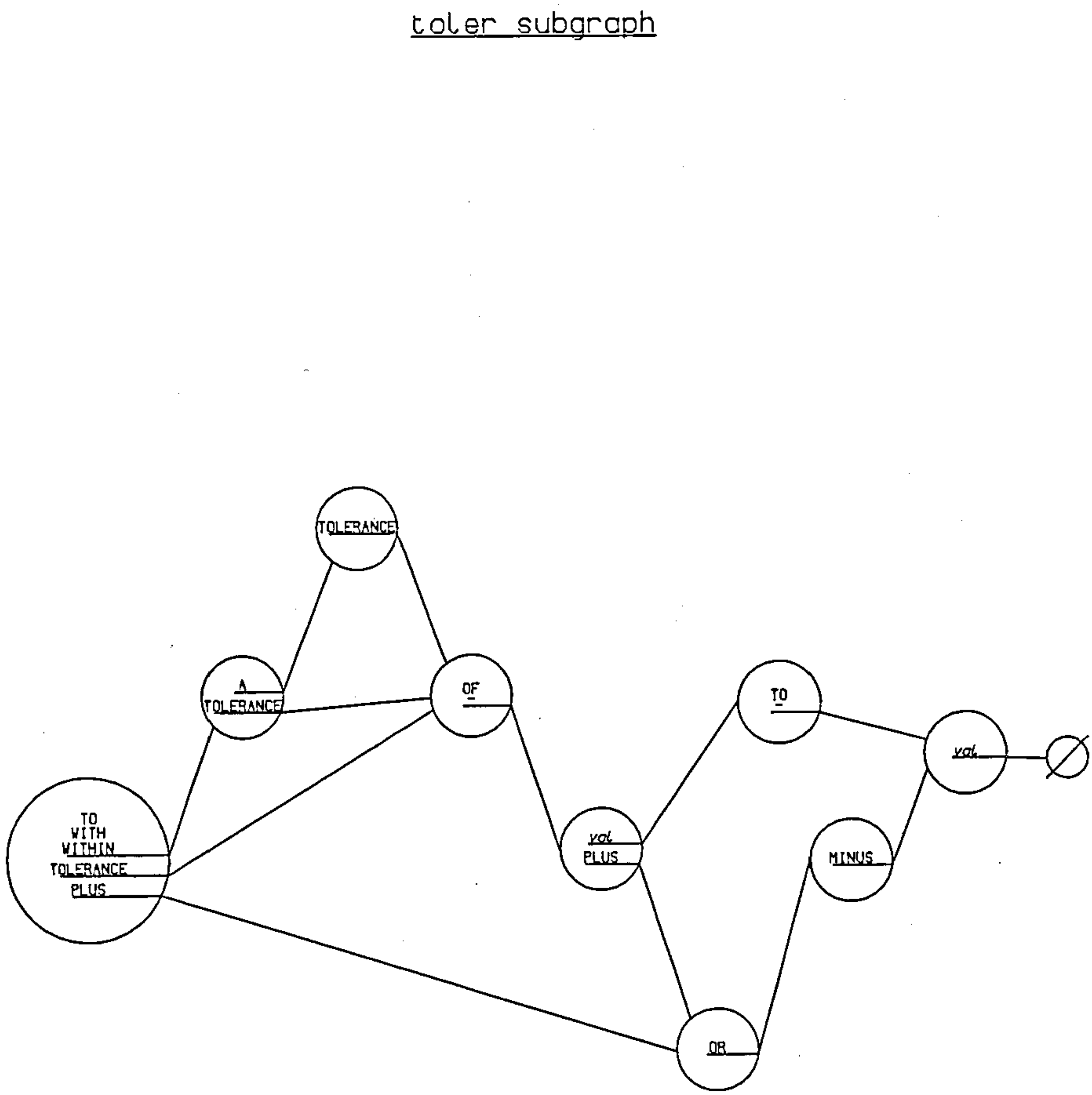


Fig 4.4 The tolerance graph

		actions	goto state
DEF TOLER			
STATE 1			
	'TO'	#	2
	'WITH'	#	2
	'WITHIN'	#	2
	'TOLERANCE'	#	4
	'PLUS'	#	6
STATE 2			
	'A'	#	3
	'TOLERANCE'	#	4
STATE 3			
	'TOLERANCE'	#	4
STATE 4			
	'OF'	#	5
	-	#	5
STATE 5			
	'PLUS'	#	6
	VAL	#	9
STATE 6			
	'OR'	#	7
STATE 7			
	'MINUS'	#	8
STATE 8			
	VAL	#	Ø
STATE 9			
	'TO'	#	8
	-	#	8
END			

Fig 4.5 Graph Input Language (GILT) form of language map

```
TO A TOLERANCE OF PLUS OR MINUS val
TO A TOLERANCE OF val TO val
TO A TOLERANCE OF val val
TO A TOLERANCE PLUS OR MINUS val
TO A TOLERANCE val TO val
TO A TOLERANCE val val
TO TOLERANCE OF PLUS OR MINUS val
TO TOLERANCE OF val TO val
TO TOLERANCE OF val val
TO TOLERANCE PLUS OR MINUS val
TO TOLERANCE val TO val
TO TOLERANCE val val
WITH A TOLERANCE OF PLUS OR MINUS val
WITH A TOLERANCE OF val TO val
WITH A TOLERANCE OF val val
WITH A TOLERANCE PLUS OR MINUS val
WITH A TOLERANCE val TO val
WITH A TOLERANCE val val
WITH TOLERANCE OF PLUS OR MINUS val
WITH TOLERANCE OF val TO val
WITH TOLERANCE OF val val
WITH TOLERANCE PLUS OR MINUS val
WITH TOLERANCE val TO val
WITH TOLERANCE val val
WITHIN A TOLERANCE OF PLUS OR MINUS val
WITHIN A TOLERANCE OF val TO val
WITHIN A TOLERANCE OF val val
WITHIN A TOLERANCE PLUS OR MINUS val
WITHIN A TOLERANCE val TO val
WITHIN A TOLERANCE val val
WITHIN TOLERANCE OF PLUS OR MINUS val
WITHIN TOLERANCE OF val TO val
WITHIN TOLERANCE OF val val
WITHIN TOLERANCE PLUS OR MINUS val
WITHIN TOLERANCE val TO val
WITHIN TOLERANCE val val
TOLERANCE OF PLUS OR MINUS val
TOLERANCE OF val TO val
TOLERANCE OF val val
TOLERANCE PLUS OR MINUS val
TOLERANCE val TO val
TOLERANCE val val
PLUS OR MINUS val
```

Fig 4.6 Potential input variations covered by the language map the command processor 'reads' an input command, atom by atom, checking against the encoded graph(s) that a complete and legitimate path is traversed. If the line is accepted, then a second pass is made to invoke ACTION's in the application program. In this manner application code is only invoked if a legitimate command is entered, although the logic of the command must still be checked within the application code.

Actions are indicated in graph input language by action numbers, following an atom. The hash symbol (#), replaces action numbers in fig 4.5 for clarity. Substitution of # for one or more positive integers enables a link to be established with corresponding code in the application program (via common blocks and computed goto statements), for the purpose of actually doing something in response to a valid command. The final integer in the chain (shown) is not an action number but the pointer to the next state. For instance, if the match in STATE 1 is 'PLUS' then a transition is made to STATE 6, where if the next atom in the command is not the word 'OR', the whole command is rejected on the grounds that a complete path cannot be traced through the tolerance graph.

Finally, a transition to state Ø means that the current graph is to be exited and thus represents a return to the calling graph, or the end of the whole map if an exit point is reached in the first or REFERENCE graph.

The process of command interpretation is of course invisible to the user who simply perceives it as a language with the variety of expression illustrated in fig 4.6.

5 THE APPLICATION OF FAMILIAR ENGINEERING LANGUAGE TO TURNING

Having already outlined the concept of familiar engineering language and briefly described the method of its application in the previous chapter, this chapter is devoted to one particular example of a language for simple turning operations. Turning was chosen as it is probably the most common engineering process and because the results could be integrated with subsequent research into other processes (78,79), thereby forming a more general system.

It was originally intended to write a familiar language graph for all turning operations, but experience gained in deriving the first graphs indicated that this was neither possible, due to limited resources, or essential in the light of evidence that many of the familiar language graphs were going to be similar.

For research purposes, the insight gained with the addition of each new graph proved subject to the law of diminishing returns. Therefore, the objective of deriving the familiar engineering language map for a sub-set of fundamental turning operations, seemed sufficent for the purpose of appraising the familiar language approach. For the want of a better expression, this sub-set of turning operations is referred to as simple turning.

The range of simple turning operations implemented in the example are as follows:-

(a) Defining the stock material prior to machining (fig 5.1).

(b) Facing across a diameter to reduce material length (fig 5.2).

(c) Linear turning along the length of an external surface to reduce material diameter (fig 5.3).

(d) Turning a step (shoulder) on an external diameter (fig 5.4).

(e) Recessing (undercutting) an external diameter (fig 5.5).

Clearly there are many other turning operations, not included in the example, that are important from an engineering point of view. Among turning operations that would be essential in a full implementation of the system are as follows:-

(a) Chamfering

(b) Parting off

(c) Taper turning

(d) Screw threading

(e) Face recessing

(f) Internal turning

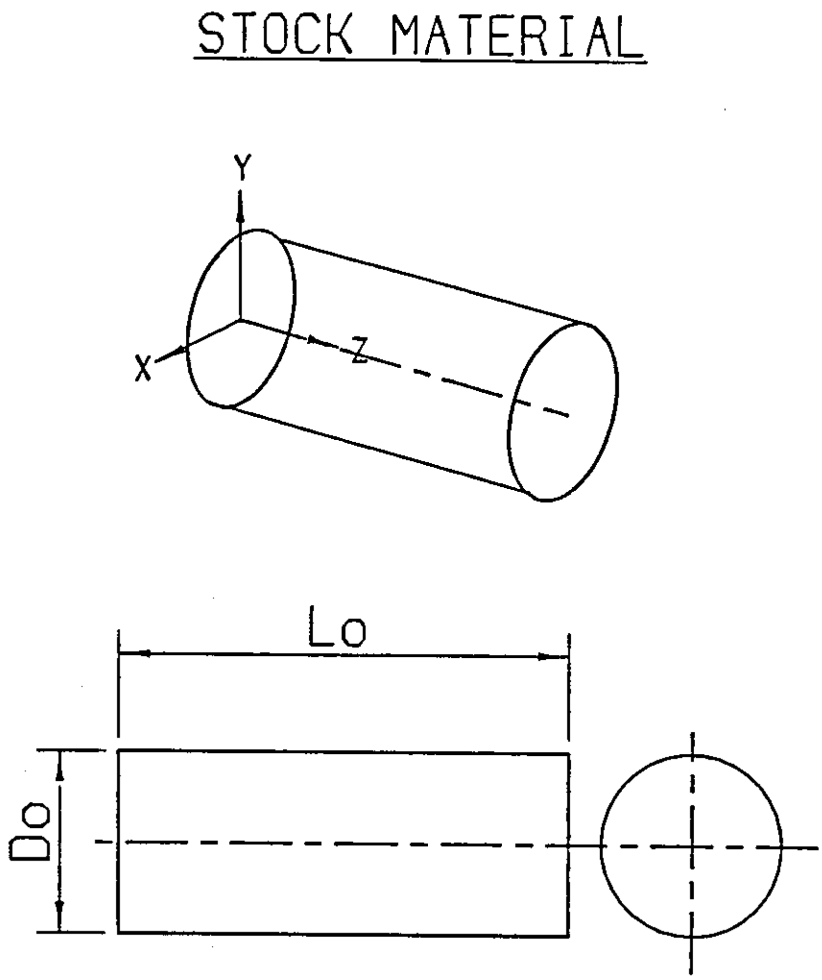


Fig 5.1 Defining stock material prior to machining

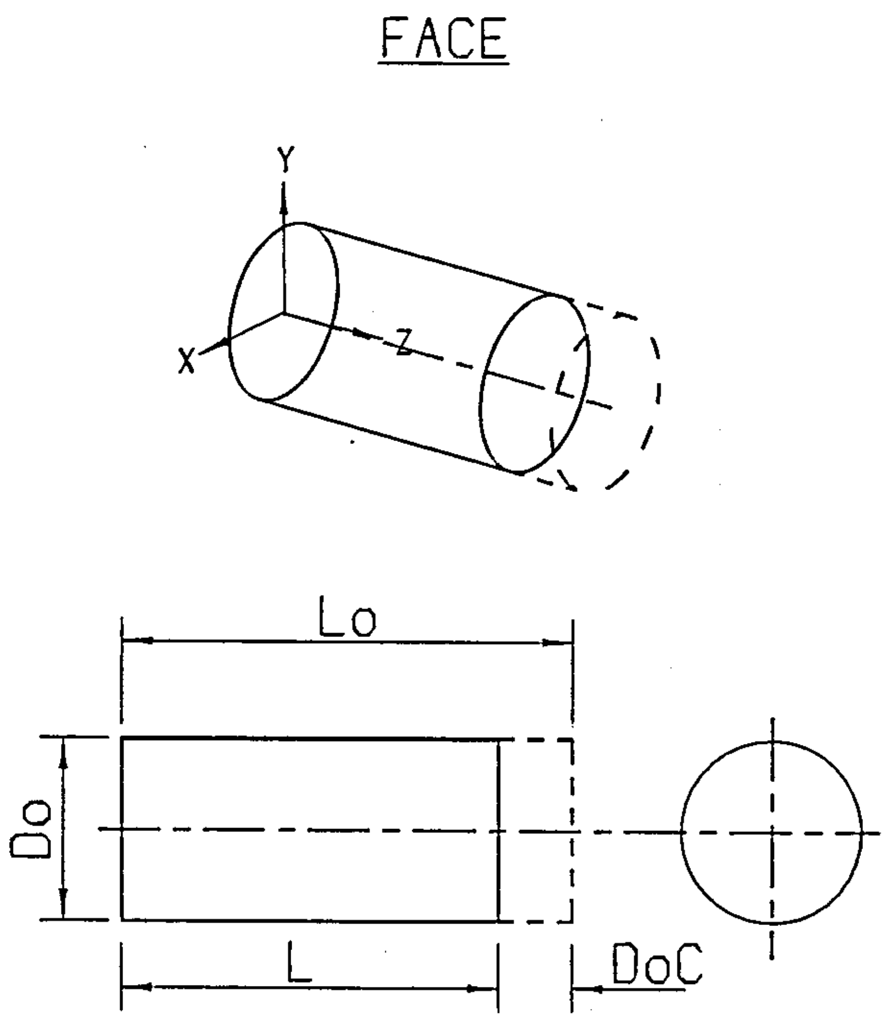


Fig 5.2 Facing across a diameter to reduce material length

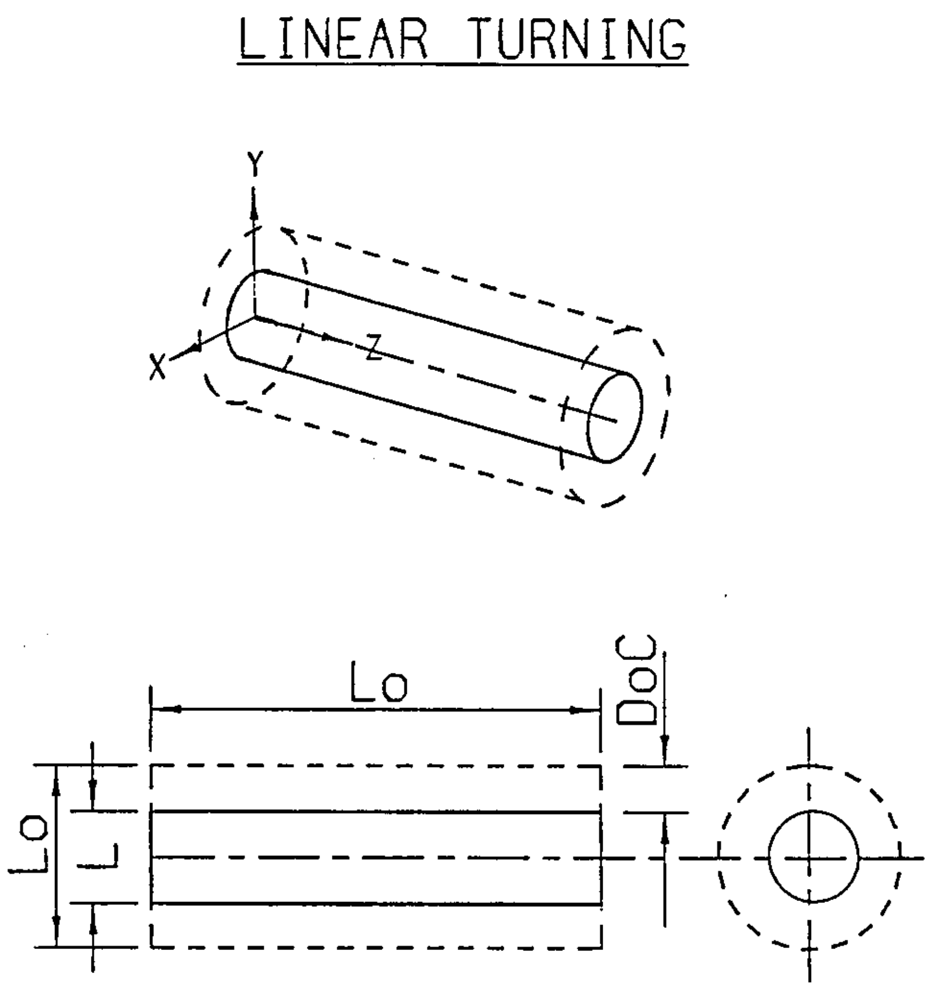


Fig 5.3 Linear turning along the length of an external surface to reduce material diameter

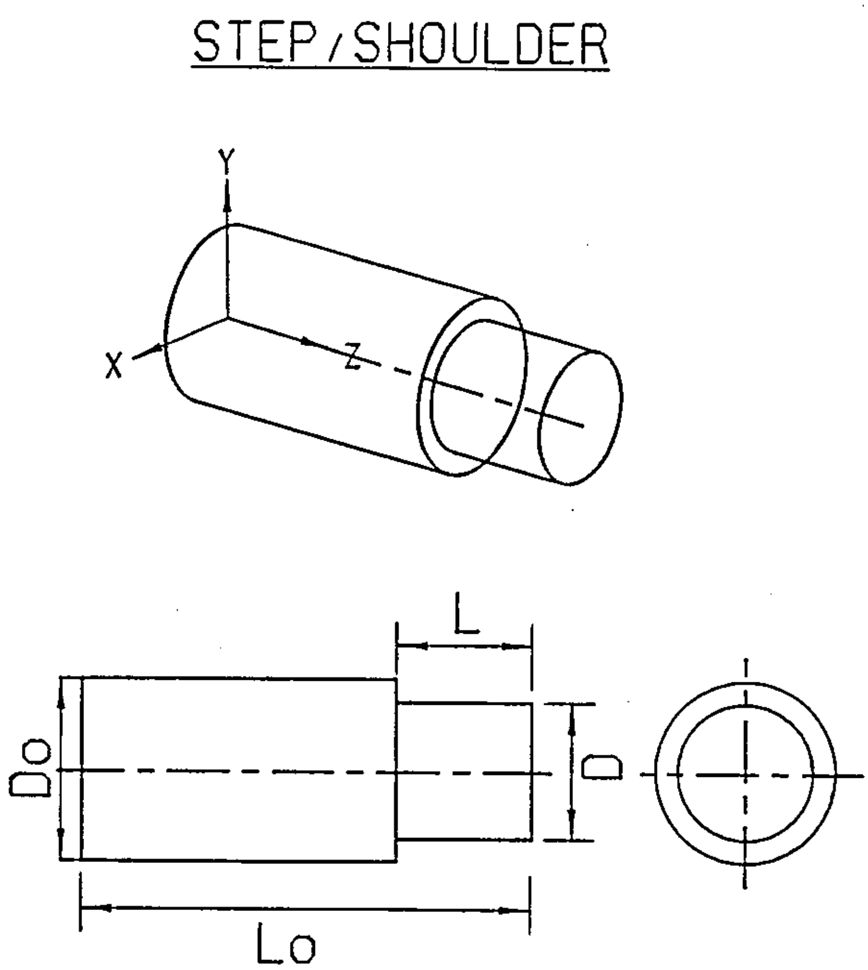


Fig 5.4 Turning a step (shoulder) on an external diameter

RECESS

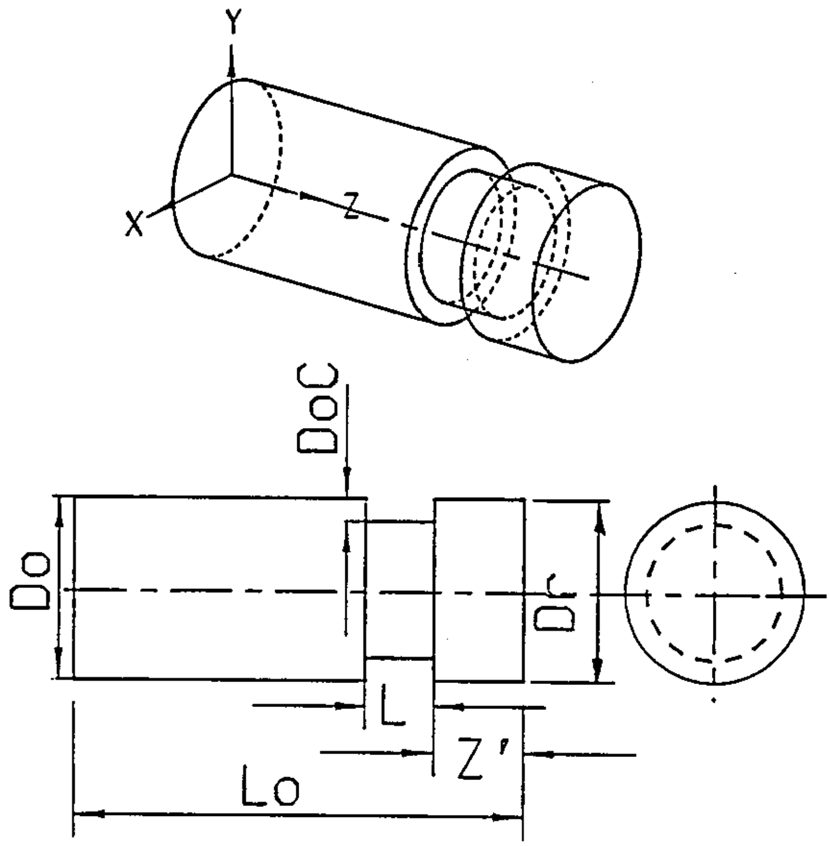


Fig 5.5 Recessing (undercutting) an external diameter

Of these, the familiar language graphs for chamfering and parting off would be relatively simple to derive, whereas taper turning and screw threading are no more difficult than the recessing operation included in the example. In figures 5.6 and 5.7, geometric specifications for incorporating taper turning and screw threading operations into the turning example are given.

Internal turning is virtually a repetition of the familiar language used in external turning operations and face recessing is obviously similar to the operation of recessing an external diameter. There are also obvious geometric similarities between these operations. A geometric specification for incoporating holes into the the turning example is given in fig 5.8.

5.1 Assumptions, constraints and conventions used in the example

Whilst flexibility in the language is of primary importance, a degree of compromise is necessary in balancing flexibility against the size of the resulting language map. Although every effort has been made to incorporate a high degree of flexibility into the simple turning example, there are limitations to the extent that this can be accomplished, not the least of which is the complexity of manually defining the

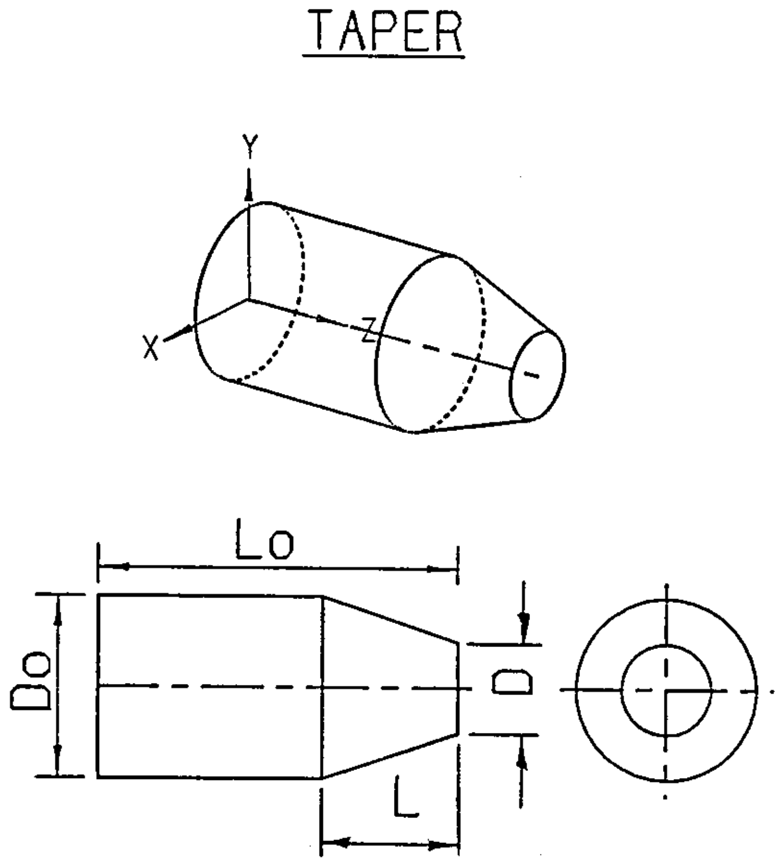
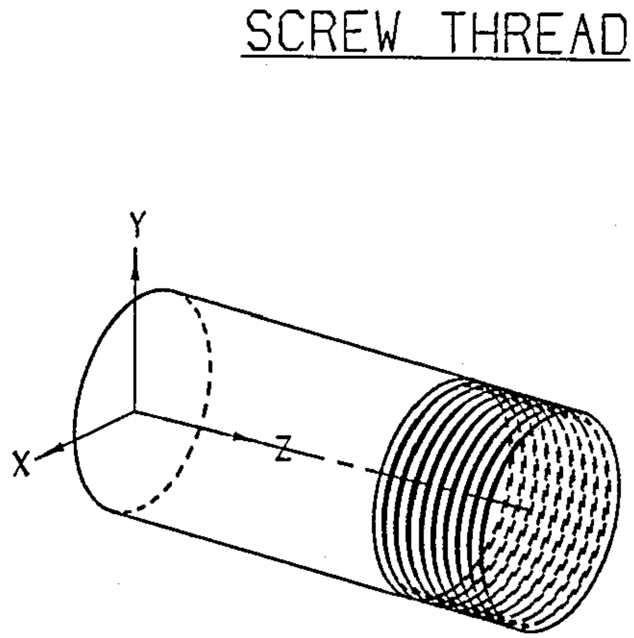


Fig 5.6 Taper turning and chamfering



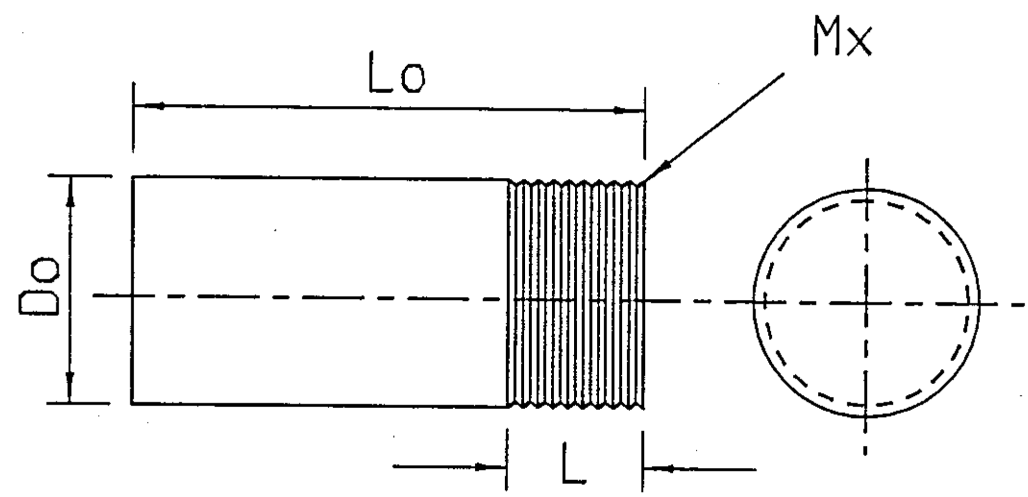


Fig 5.7 Turning a screw thread

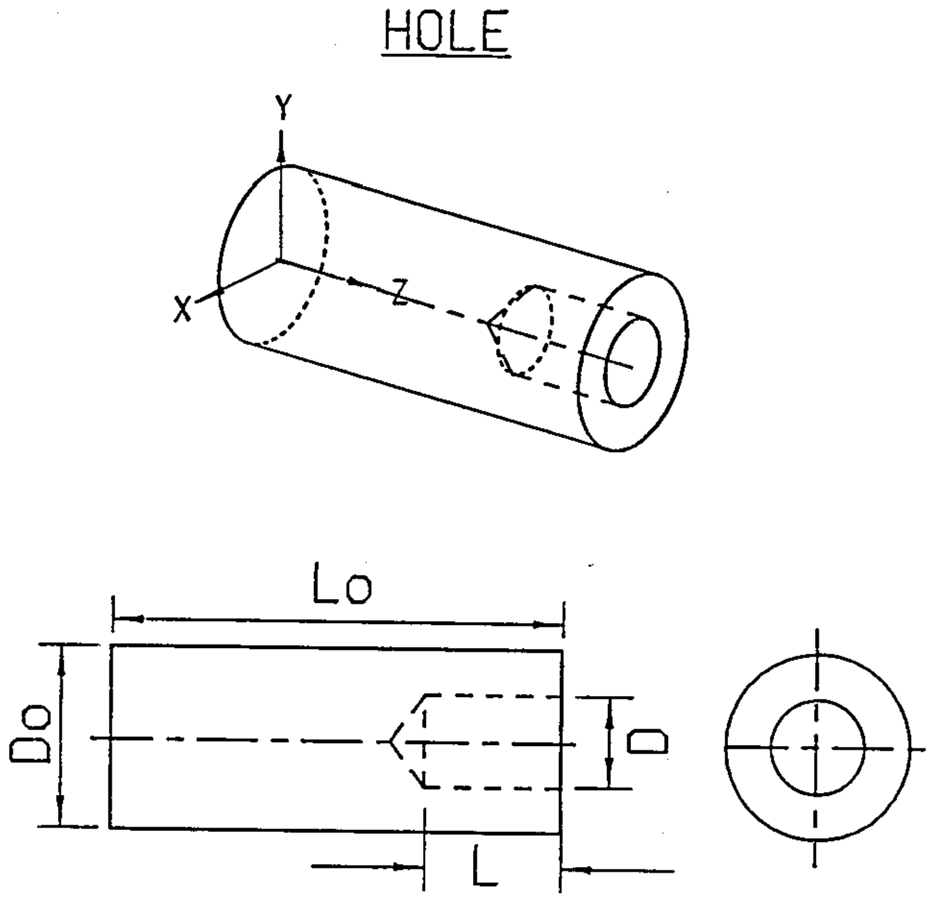


Fig 5.8 Drilling a hole

language structure. Another limitation is the potential growth of ambiguity in the language, indicating that it is probably better to customize several versions of the language according to need (eg regional/company variations), rather than attempt to develop a single universal familiar language. A virtue of the command processor approach is the ease with which a basic version of familiar engineering language can be customized.

If complete freedom of expression is allowed, subject to the rule that 'legal' commands are unambiguous, even with a customized vocabulary, the number of permissible constructions is enormous. This combinatorial explosion is a problem, but if standardization is employed, the problem is containable. It is apparent that a degree of standardization is desirable if ambiguous and redundant commands are to be avoided. Standardization is important for other reasons too, for example, the following command although unambiguous and without redundant information, is clearly unorthodox in construction:-

TO DIAMETER 20.0 AND WIDTH 5.5 PLUS OR MINUS 0.05
AT 10.0 FROM THE END TURN A RECESS IN THE BAR

The construction is much more logical and conventional if the nature of the operation is specified first:-

RECESS THE BAR TO DIAMETER 20.0 AND WIDTH 5.5

PLUS OR MINUS 0.05 AT 10.0 FROM THE END

The effect of specifying the nature of the operation before any supporting information also dramatically reduces the number of possible combinations in the resulting familiar language graph.

Clearly absolute flexibility is undesirable if the result is to encourage convoluted commands, therefore it is also important to standardize. Consequently, in the example of familiar language for simple turning, the practice of specifying supporting information before indicating the nature of an operation is not allowed. Similar judgements about choice of semantics have been made in the example language map which would not suit every application, therefore, the ease with which the semantics of the language map can be modified is a considerable virtue.

The degree of standardization employed in the example is quite elementary and this is substantiated by the number of command permutations that can be generated from the example language map. Considerable variety of construction is still possible, commands can be either long and verbose or short and succinct according to taste; for example:-

FACE THE BAR DOWN TO A LENGTH OF 100.0

FACE TO 100.0

Both of which are interpreted as equivalent commands. In the case of input using the shorter version, the potential exists to output a longer version of the command for clarity, which would be more appropriate in process planning, although this facility is not explicitly implemented in the example.

The reasoning behind this being that as users of a system become more familiar with it, the speed with which it can be used may be significantly improved by using a shortened form of command input. The tedium of repetitively typing long commands for non-touch-typists is also allieviated. However, from the output point of view there is little to be gained by using shortened and therefore less explicit commands, as for example in a process plan, so long versions of the same command could be output instead.

In the interests of flexibility, considerable freedom is allowed in the manner and order in which dimensional information can be expressed in support of a command. Naturally, the nature of a command allowing this amount of freedom is going to be verbose if ambiguity is to be avoided. Therefore, in order to shorten the command format significantly, a default sequence of dimensioning must be employed. This has the effect of allowing very short forms of commands that may appear ambiguous, because examination of the language graphs alone cannot reveal the fact that defaults are implicit in the application program. The previous example of a shortened form of the 'face' command cannot be ambiguous because only one dimension is necessary to describe the facing operation (length), but now consider the equivalent problem in the case of recessing:-

RECESS THE BAR TO DIMENSIONS OF 3Ø.Ø DIAMETER AND WIDTH 1Ø.Ø AT A DISTANCE OF 2Ø.5 FROM THE END

RECESS 3Ø.Ø 1Ø.Ø 2Ø.Ø5

Again both of these are considered to be equivalent commands, but the shorter version can only be interpreted by applying a default order of dimensioning. The defaults have been chosen arbitrarily so that a consistent pattern can be observed between graphs of different types thus:-

Diameter...Width/Length

Diameter...Width/Length...Location

In graphs requiring two dimensions to specify the operation, for example 'stock', the first of the above is used, whereas in 'reces', which requires three dimensions, the second is used. Note that although the terms width and length can be considered as synonymous, the default presumes that diameters and not radii are specified. Clearly, short versions of commands like this are ambiguous in appearance (not to the system), but are desirable if the needs of experienced as well as novice or casual users are to be met. However, where such shortened commands are likely to be used, it is strongly recommended that a longer version of the command be used for output or as feedback to the user.

Of course further shortening of commands is still possible using abbreviations in the language graphs. This technique was explained briefly in the previous chapter but has not been attempted in the example because the substitution of abbreviations presents little technical challenge. A further degree of customization is possible by the user, who may create a macro file of abbreviated commands himself, a facility that is independent of language construction.

Returning to the point about default assumptions, it is possible to specify certain dimensions explicitly and leave others to default so long as a logical command is entered, because this possibility has been considered in the language graphs. For example:-

> TURN AN UNDERCUT IN THE OUTSIDE DIAMETER OF THE BAR AT A DISTANCE OF 2Ø.5 FROM THE FREE END WITH DIMENSIONS OF WIDTH 1Ø.Ø TO A TOLERANCE OF PLUS OR MINUS Ø.1 AND A DIAMETER OF 5Ø.Ø

UNDERCUT AT 2Ø.Ø5 WIDTH 1Ø.Ø PLUS OR MINUS Ø.1 BY 5Ø.Ø

Notice here that because the ordering of dimensions does not completely follow the default sequence, additional information must be given, but once the meaning of the command is clear the user can rely on a logical default. An alternative construction relying further on the defaults might be:-

UNDERCUT AT 2Ø.Ø5 5Ø.Ø 1Ø.Ø PLUS OR MINUS Ø.1

Shorter alternatives to the tolerancing part have been avoided to enhance distinction between numeric and alphabetic text, although the potential for abbreviation is present.

A simple convention is introduced in the example to reference diameters on a component, as a method of specifying dimensions and tolerances relative to local datums. Although the use of appropriate names for particular features on a component is probably the common engineering approach, it can be ambiguous. Instead, the simple numeric convention shown in fig 5.9 is used whereby diameters are automatically renumbered from the tailstock end of the workpiece after every operation. An example of a command containing such a reference is:-

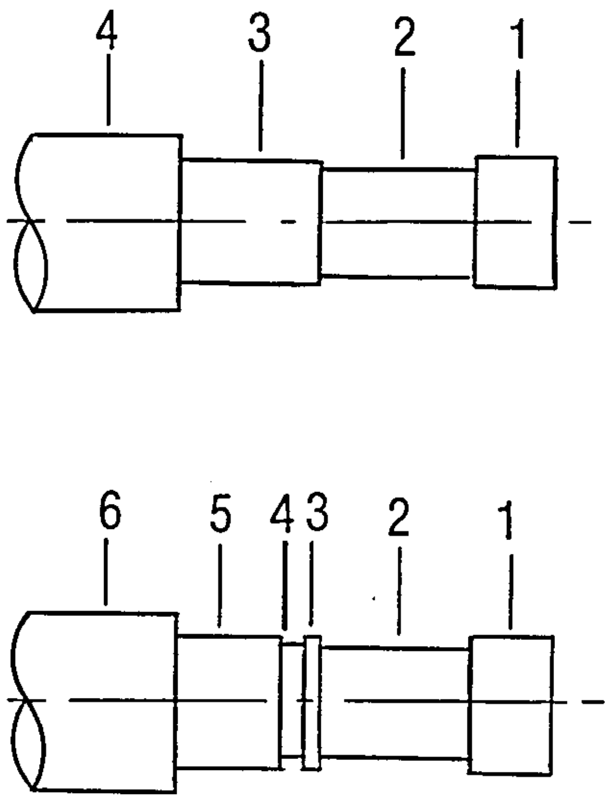


Fig 5.9 The convention used to reference diameters

TURN A STEP IN DIAMETER 3 TO SIZE 20.2 WIDE BY RADIUS 30

An alternative approach is used in the process planning system PROPLAN, described in chapter three, where diameters are referenced directly using their associated dimensions (see figs 3.8 and 3.9). However, it is easy to see how this approach could become unwieldy for complex components. The risk of ambiguity is also present if two or more diameters of the same size occur on one component.

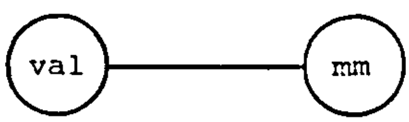
Some other details that the reader might expect to find in the familiar engineering language map for simple turning have been deliberately omitted. Three main reasons exist for this:-

(a) Standardization
(b) Simplification
(c) Streamlining

Examples of the application of standardization to encourage good practice and efficiency have already been mentioned. Simplification of minor aspects of the familiar language map has enabled more progress to be made on more important aspects. By streamlining the familiar language map, obscure or redundant commands have been eliminated.

An example of simplification is the graph for defining stock material, where for geometric purposes the type of material (mild steel, brass etc) is irrelevent, therefore, the capacity to specify it has been omitted. It would be comparatively simple, although time consuming, to enlarge the stock graph to include a suitable range of material types.

Some incidences of simplification occur globally, rather than in single graphs. For instance, the example language map does not incorporate the postfix notation for metric dimensioning (mm), thus reducing the total number of states in the language map. The notation could be added by defining an extra state following every call to 'val', tedious but not difficult thus:-



Similarly, no attempt has been made to exhaustively define all of the potential engineering synonyms, instead a few common terms have been selected. It is a technically simple but time consuming task to add further synonyms, though there is scope for automating this procedure.

Engineering attributes like surface finish, have not been included in the example on the basis that the variety of surface finish typically used in turning is very limited, which combined with prudent tolerance definition, makes the specification of surface finish redundant.

No distinction is made between rough and finish turning operations in the example familiar language map for simple turning, in fact neither of the words have been included in the vocabulary. Although the distinction between roughing and finishing operations is common practise in industry and therefore appears to warrant inclusion, they are in reality, sub-operations of a single turning operation. If due consideration is given to tolerancing, then the terms roughing and finishing, which are ambiguous anyway, become redundant.

The roughing/finishing dilemma, highlights the conflict between common engineering terminology and 'idealized' terminology, which has to be resolved during the derivation of familiar engineering language. While it is important to retain familiarity in the choice of terminology, it is also important to encourage the standardization of correct terminology. A compromise solution might be to allow the terms roughing/finishing without attaching any meaning to them, but this option was not preferred.

5.2 Anatomy of the familiar language map for simple turning

The only satisfactory way to appreciate the nature of the familiar language map derived for simple turning is to study it, therefore the complete set of graphs have been included in this chapter.

In total, there are ten seperate language graphs constituting the familiar language map for simple turning. Ideally there should only be six but to reduce the recessing graph to a manageable size it has been split into four seperate graphs. The graph names are kept to a maximum of five characters (a constraint of GILT) as follows:-

(1) refer - the graph from which all others are referenced.

(2) stock - the graph for the definition of stock material.

(3) face - the graph of operations reducing material length

(4) linea - the graph of operations reducing material diameter.

(5) step - the graph of step/shoulder turning operations.

(6) reces - the graph splitting the graph of recessing/undercutting operations into three graphs.

(7) recsa - recessing subgraph a

(8) recsb - recessing subgraph b

(9) recsc - recessing subgraph c

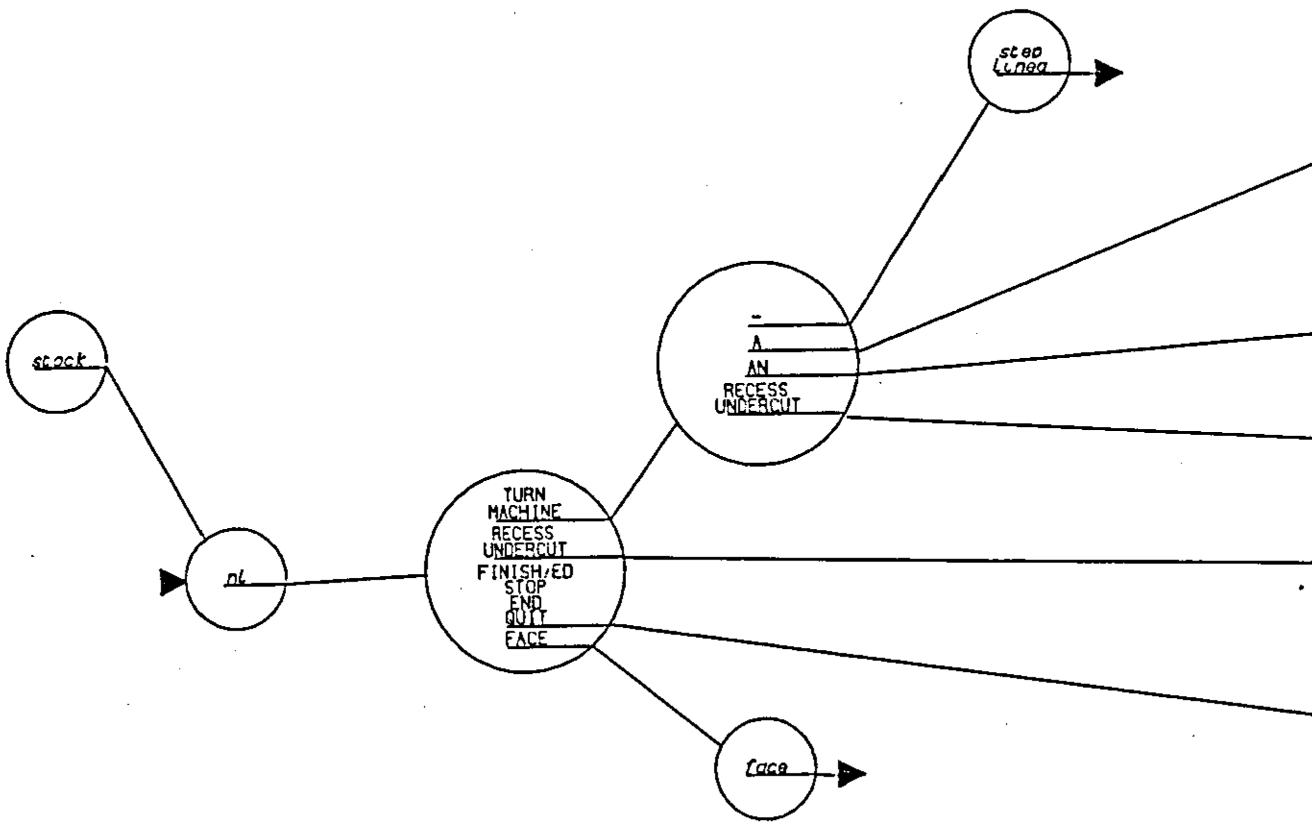
(10) toler - subgraph for specification of tolerances.

In fig 5.10 for convenience, is a quick reference guide to the symbols and notation used in the familiar language graphs, summarizing the explanation given in full in the previous chapter.

○	state
CAPITALS	atoms
italics	call to another graph or subgraph
Atom	transitions - applies to all words above line
▶	continuation markers used to link two halves of a graph
-	null transition
val	internal GILT graph that captures a number
nl	indicates end of command - new line or return

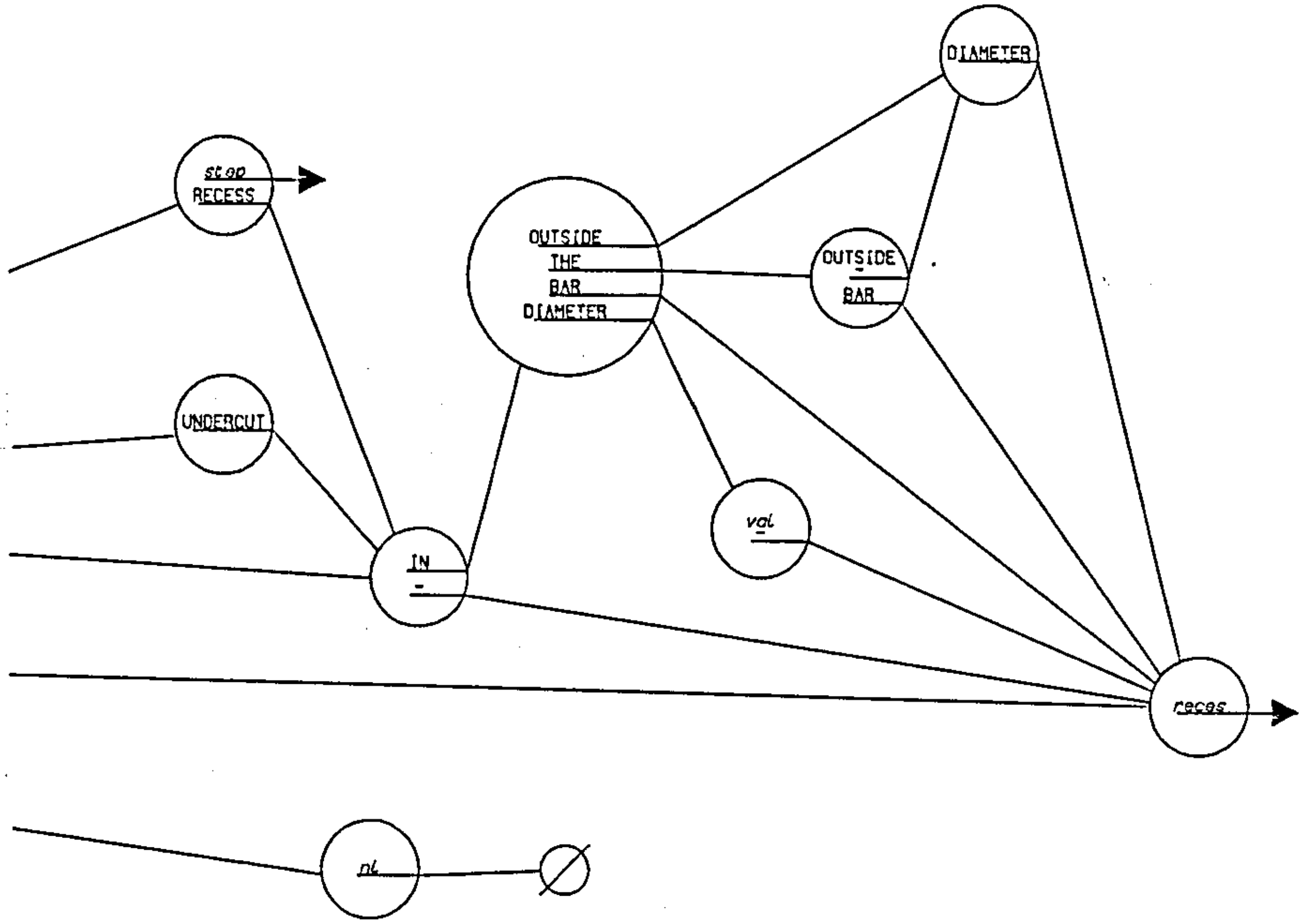
Fig 5.10 Key to symbols used in graphs

refer graph



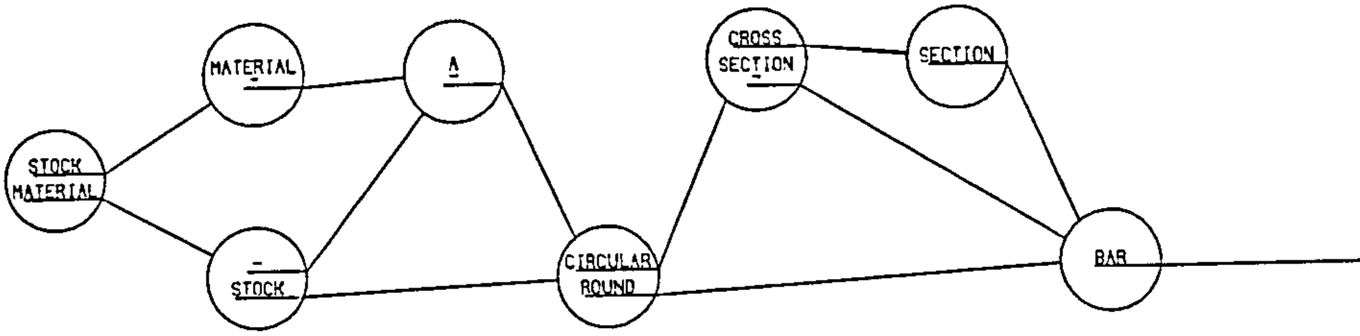
step
linea
-
A
AN
RECESS
UNDERCUT
stock
TURN
MACHINE
RECESS
UNDERCUT
FINISH/ED
STOP
END
QUIT
FACE
nl
face

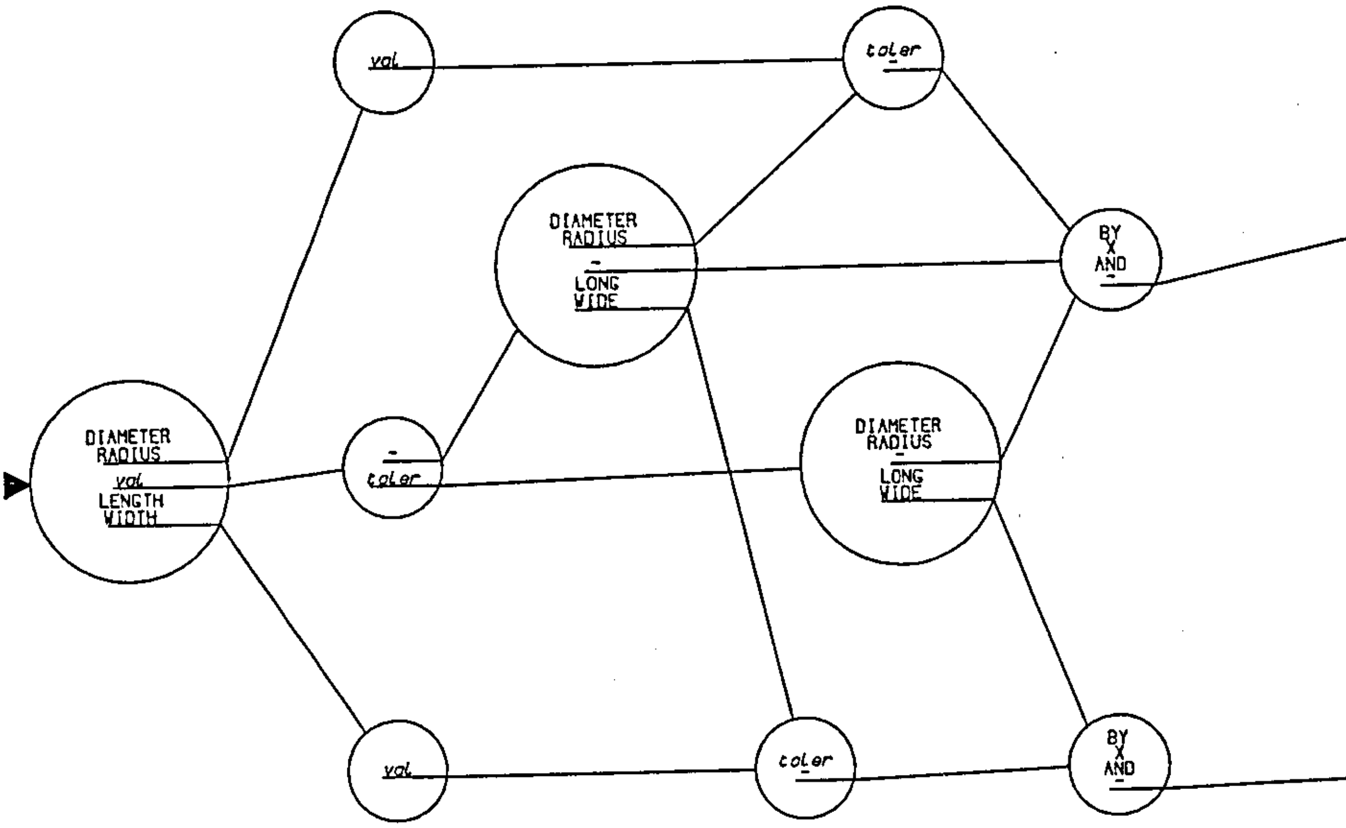
refer graph



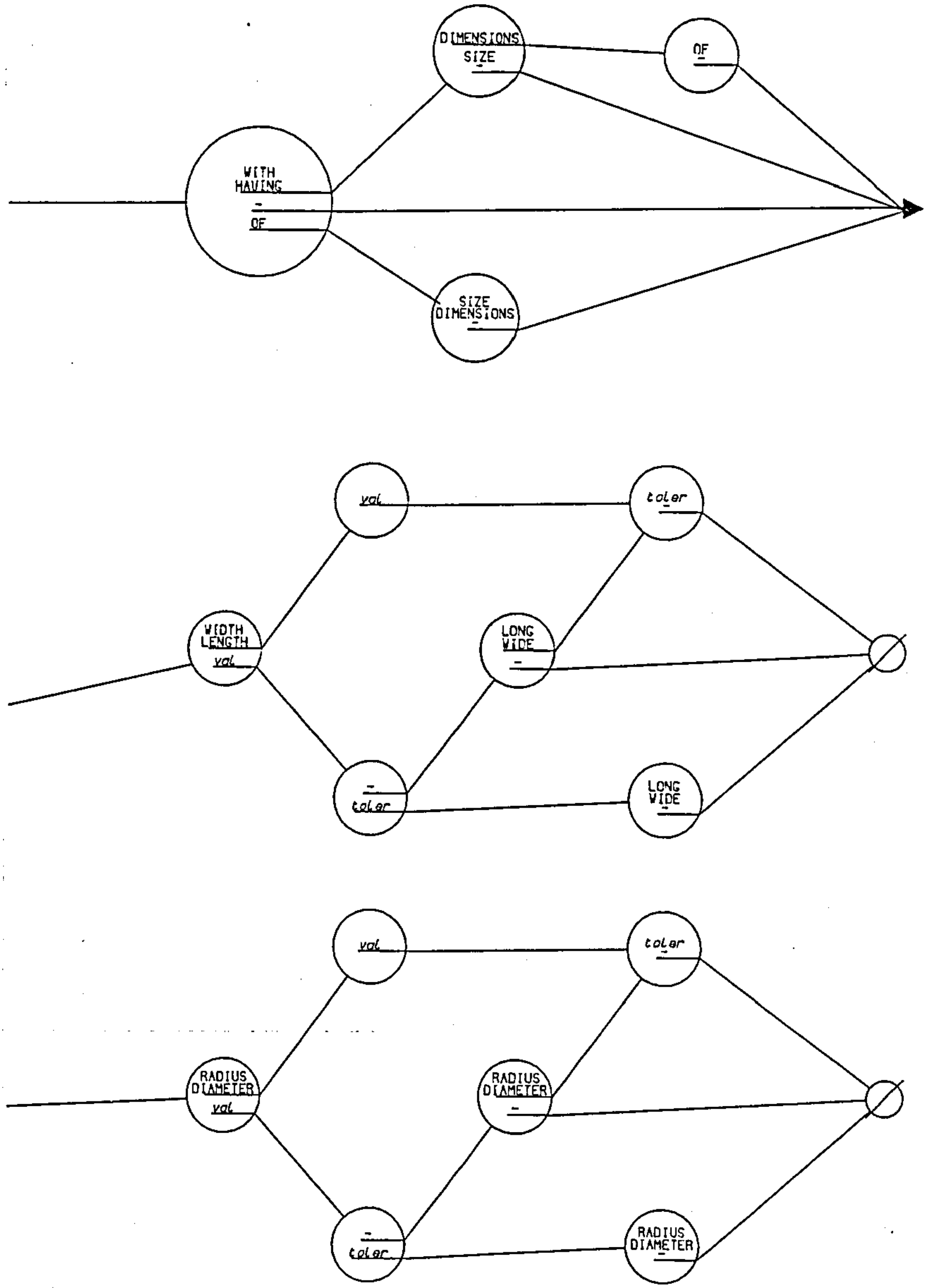
stop
RECESS
UNDERCUT
IN
-
OUTSIDE
THE
BAR
DIAMETER
DIAMETER
OUTSIDE
-
BAR
val
-
reces
nl

stock graph

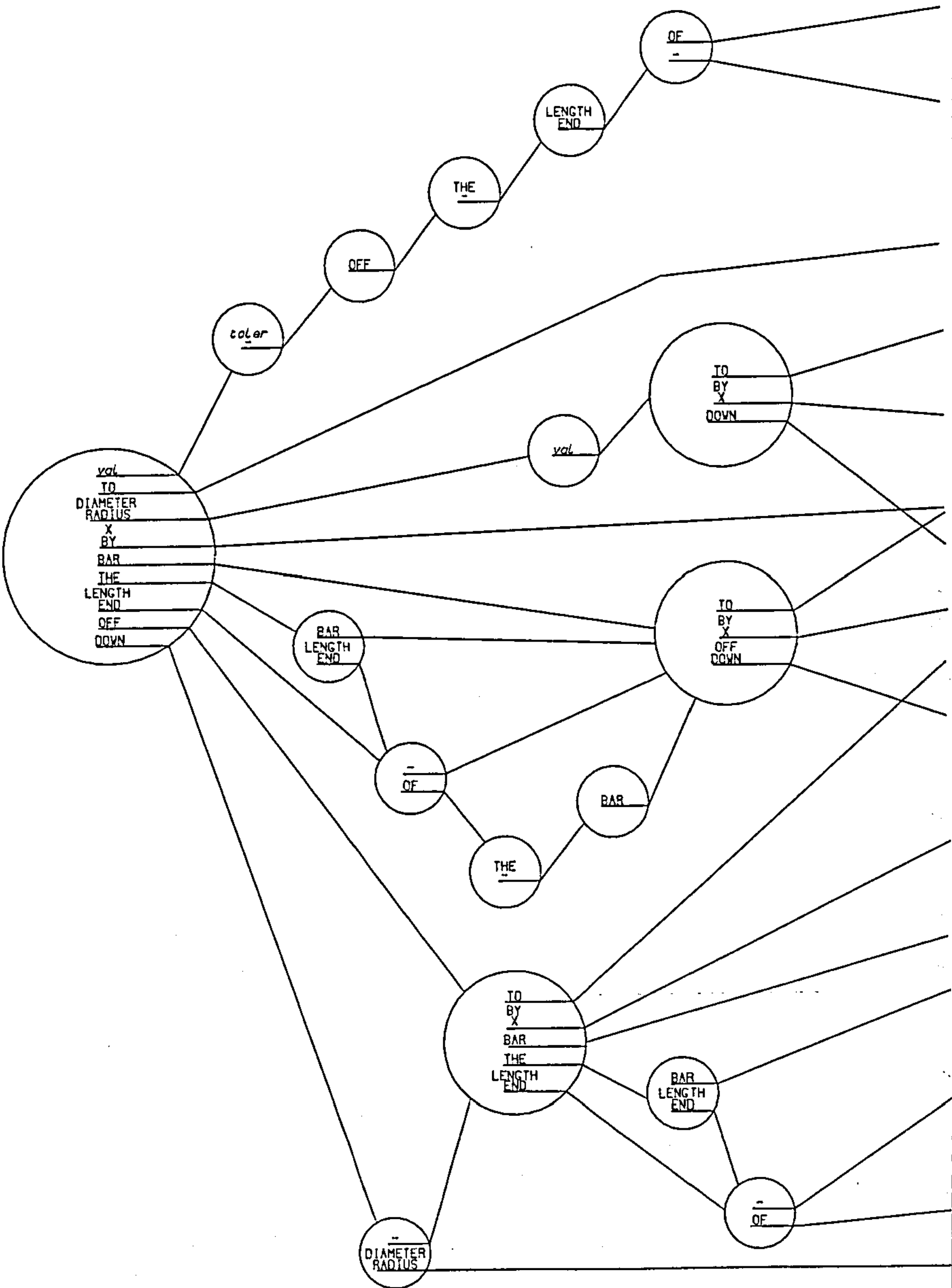

STOCK
MATERIAL
MATERIAL
A
STOCK
CIRCULAR
ROUND
CROSS
SECTION
SECTION
BAR


val
toler
DIAMETER
RADIUS
LONG
WIDE
BY
X
AND
DIAMETER
RADIUS
val
LENGTH
WIDTH
toler
DIAMETER
RADIUS
LONG
WIDE
val
toler
BY
X
AND

stock graph

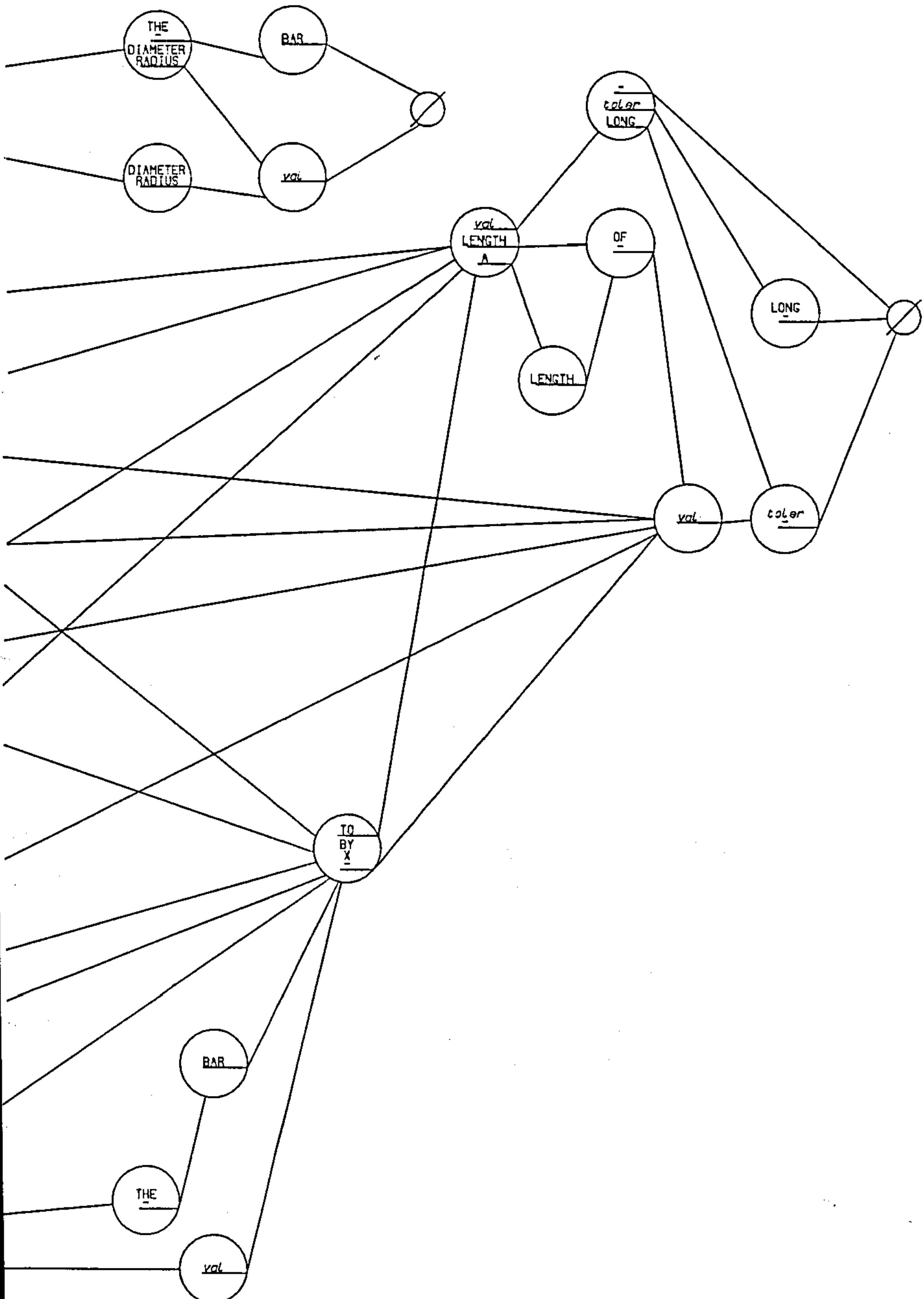

DIMENSIONS
SIZE
OF
WITH
HAVING
OF
SIZE
DIMENSIONS
val
toler
WIDTH
LENGTH
val
LONG
WIDE
toler
LONG
WIDE
val
toler
RADIUS
DIAMETER
val
RADIUS
DIAMETER
toler
RADIUS
DIAMETER

face graph

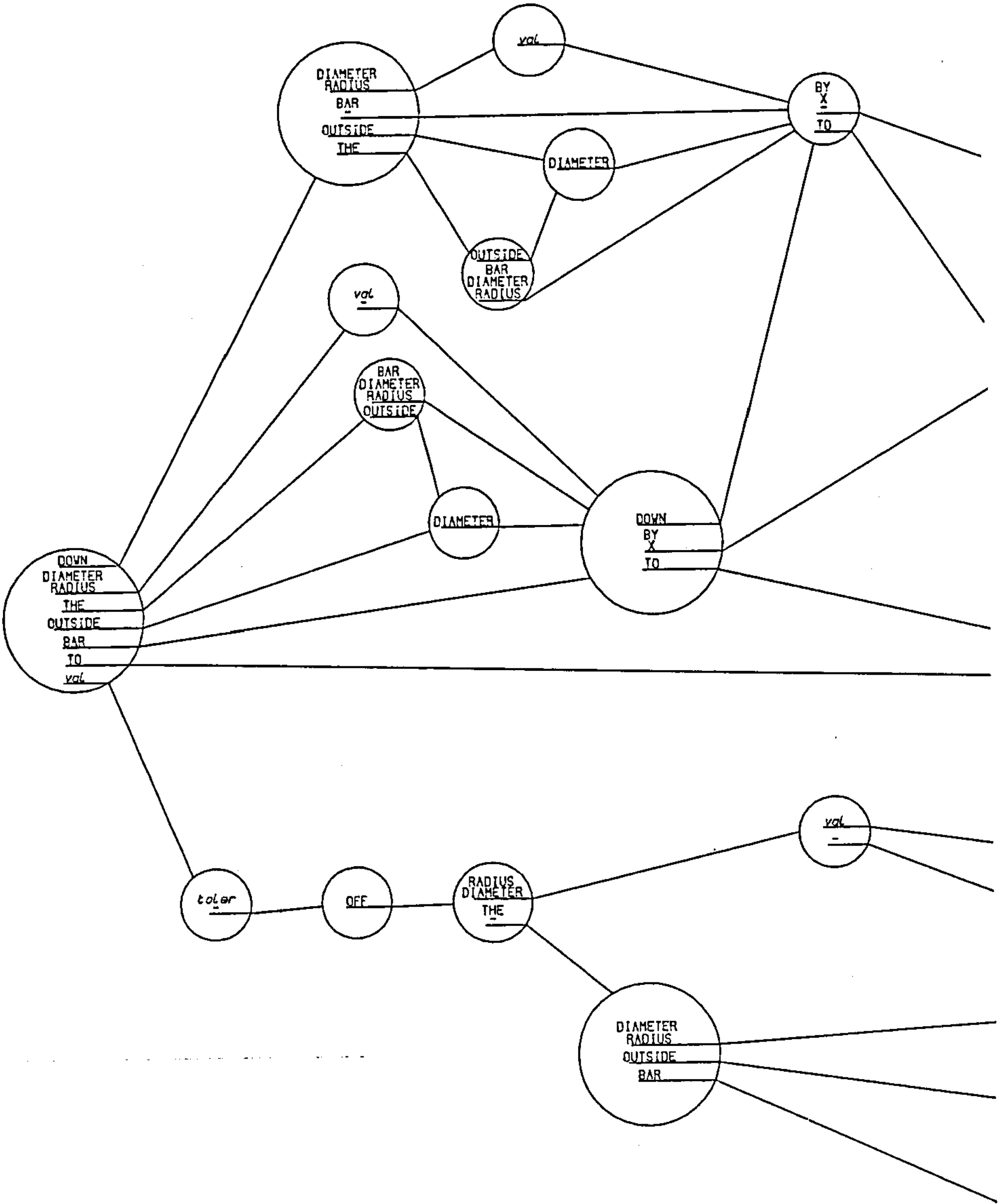


OF
-
LENGTH
END
THE
-
OFF
toler
-
val
TO
DIAMETER
RADIUS
X
BY
BAR
THE
LENGTH
END
OFF
DOWN
val
TO
BY
X
DOWN
BAR
LENGTH
END
TO
BY
X
OFF
DOWN
-
OF
BAR
THE
-
TO
BY
X
BAR
THE
LENGTH
END
BAR
LENGTH
END
-
OF
-
DIAMETER
RADIUS

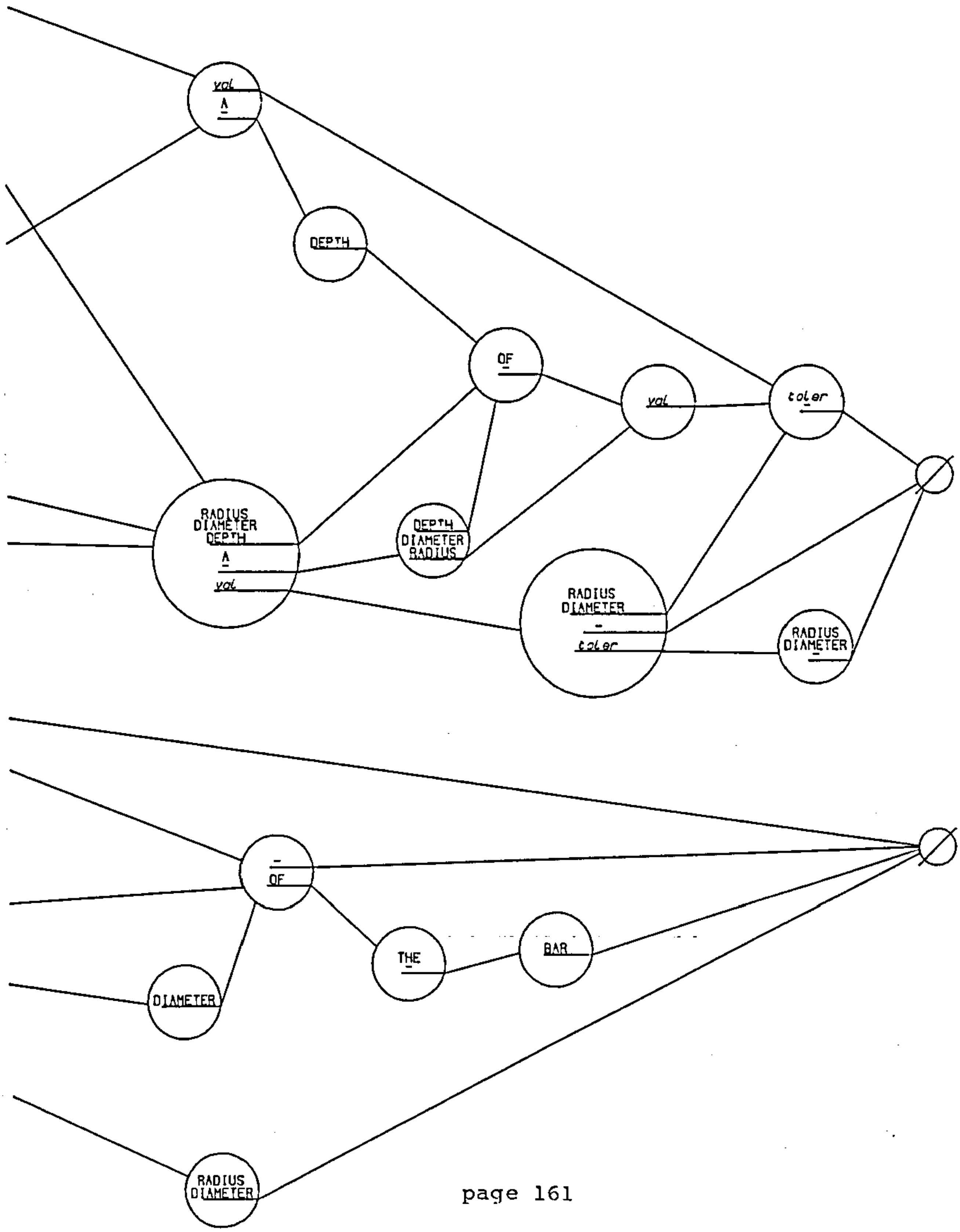
face graph


THE
DIAMETER
RADIUS
BAR
DIAMETER
RADIUS
val
color
LONG
val
LENGTH
A
OF
LENGTH
LONG
val
color
TO
BY
X
BAR
THE
val

Lunea graph

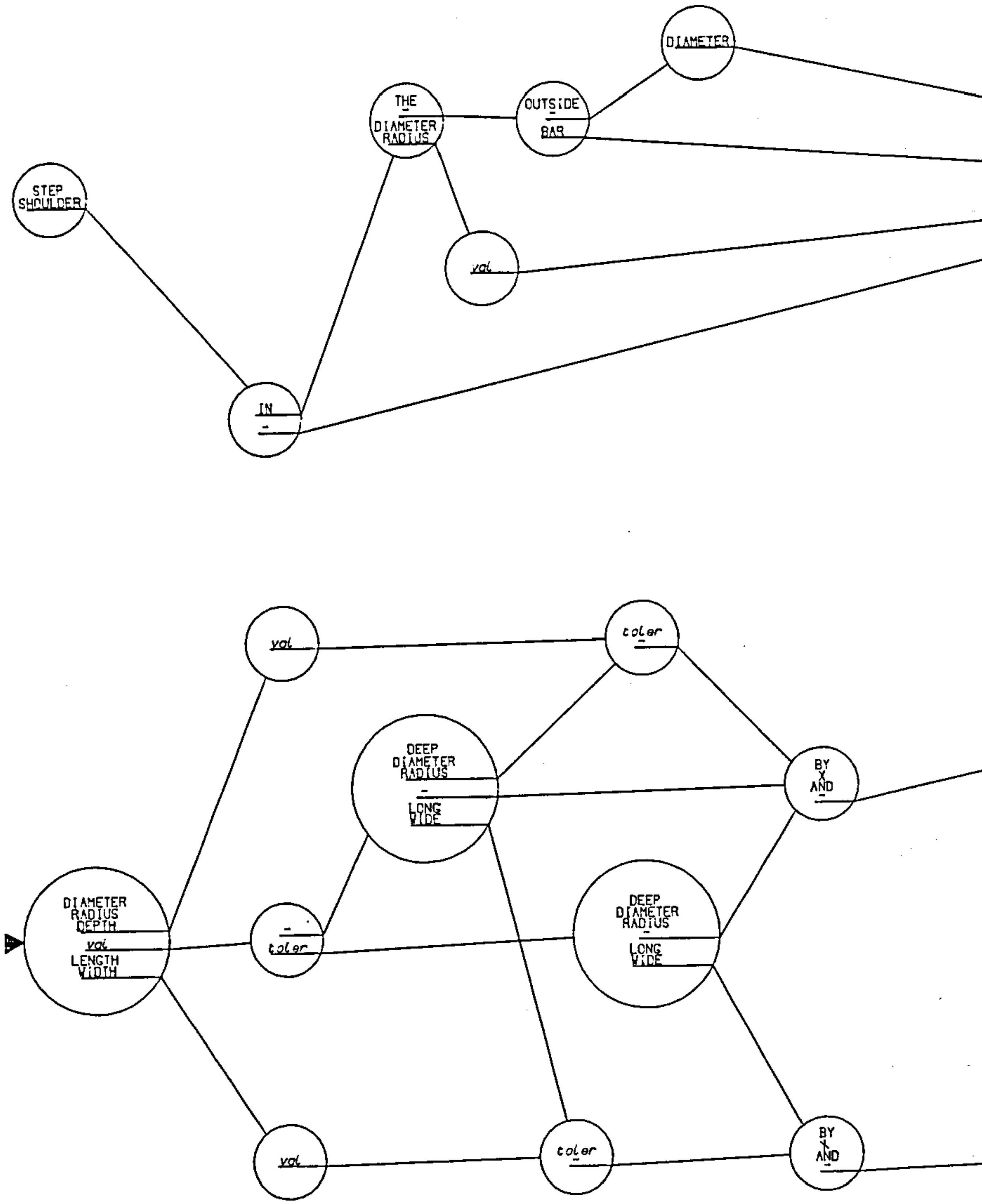


val
DIAMETER
RADIUS
BAR
OUTSIDE
THE
BY
X
TO
DIAMETER
OUTSIDE
BAR
DIAMETER
RADIUS
val
BAR
DIAMETER
RADIUS
OUTSIDE
DIAMETER
DOWN
BY
X
TO
DOWN
DIAMETER
RADIUS
THE
OUTSIDE
BAR
TO
val
val
toler
OFF
RADIUS
DIAMETER
THE
DIAMETER
RADIUS
OUTSIDE
BAR



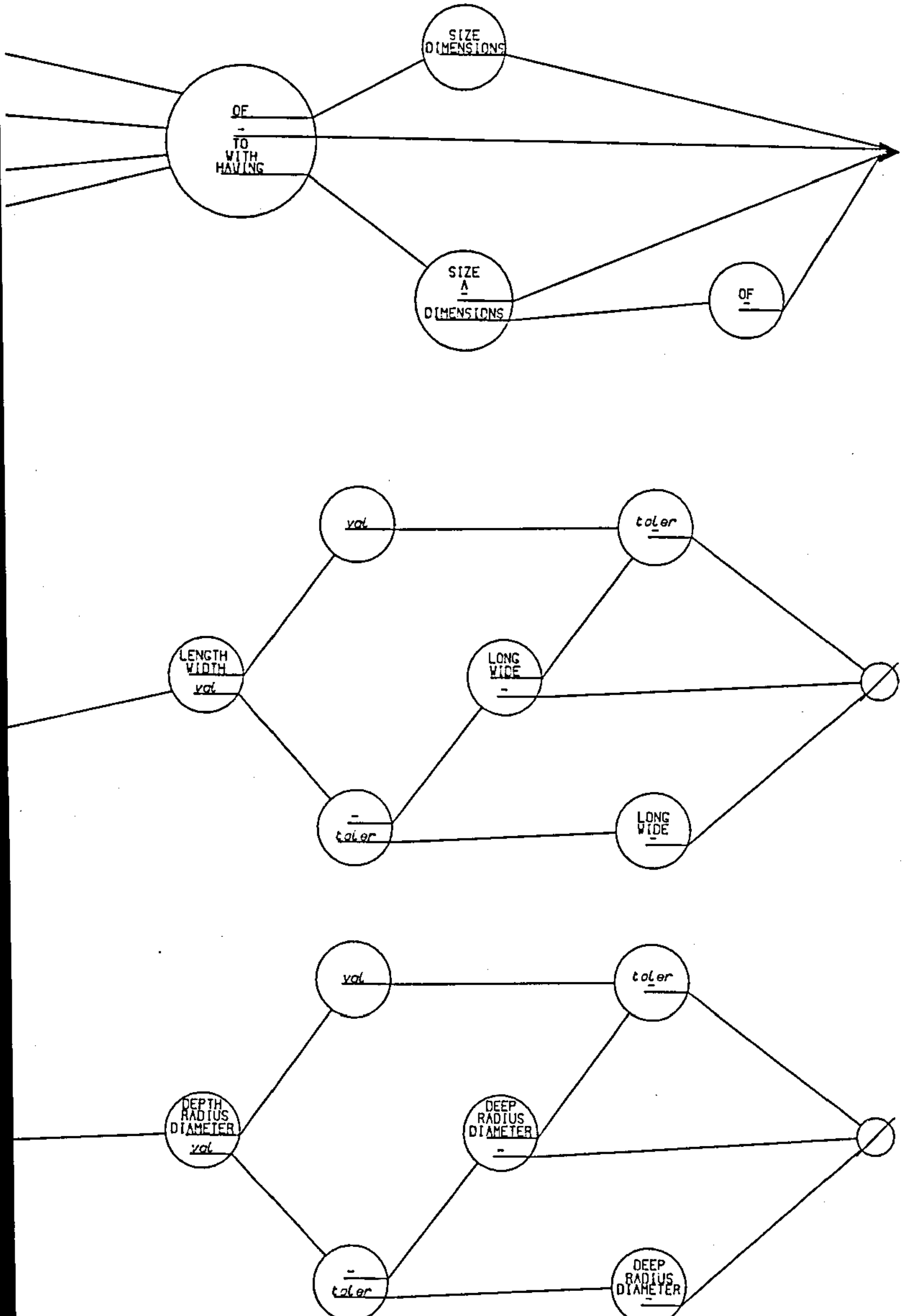
val
A
DEPTH
OF
val
toler
RADIUS
DIAMETER
DEPTH
A
val
DEPTH
DIAMETER
RADIUS
RADIUS
DIAMETER
toler
RADIUS
DIAMETER
OF
THE
BAR
DIAMETER
RADIUS
DIAMETER

step graph

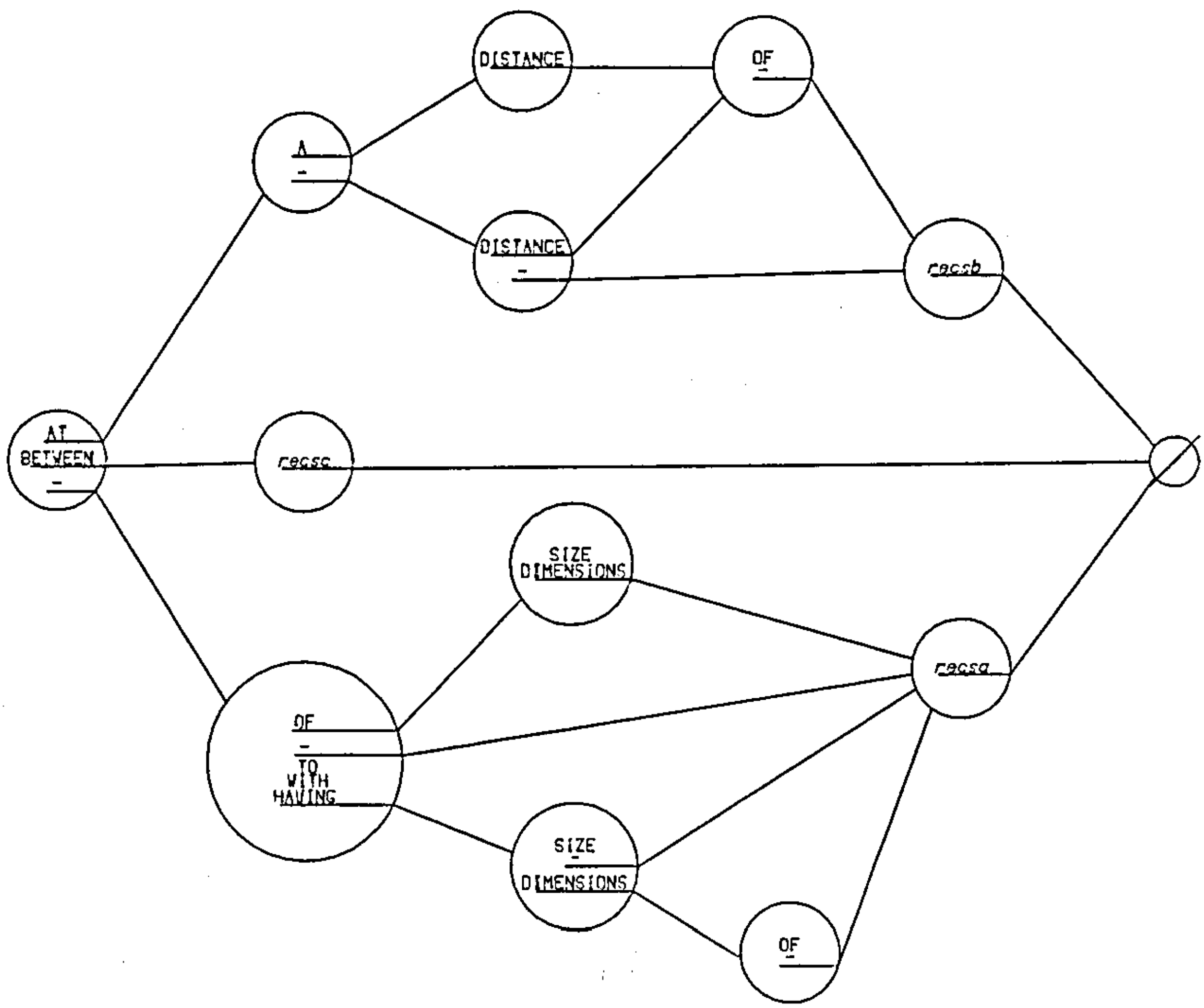


STEP
SHOULDER
IN
THE
DIAMETER
RADIUS
OUTSIDE
BAR
DIAMETER
val
DIAMETER
RADIUS
DEPTH
val
LENGTH
WIDTH
val
toler
DEEP
DIAMETER
RADIUS
LONG
WIDE
toler
BY
X
AND
DEEP
DIAMETER
RADIUS
LONG
WIDE
val
toler
BY
X
AND

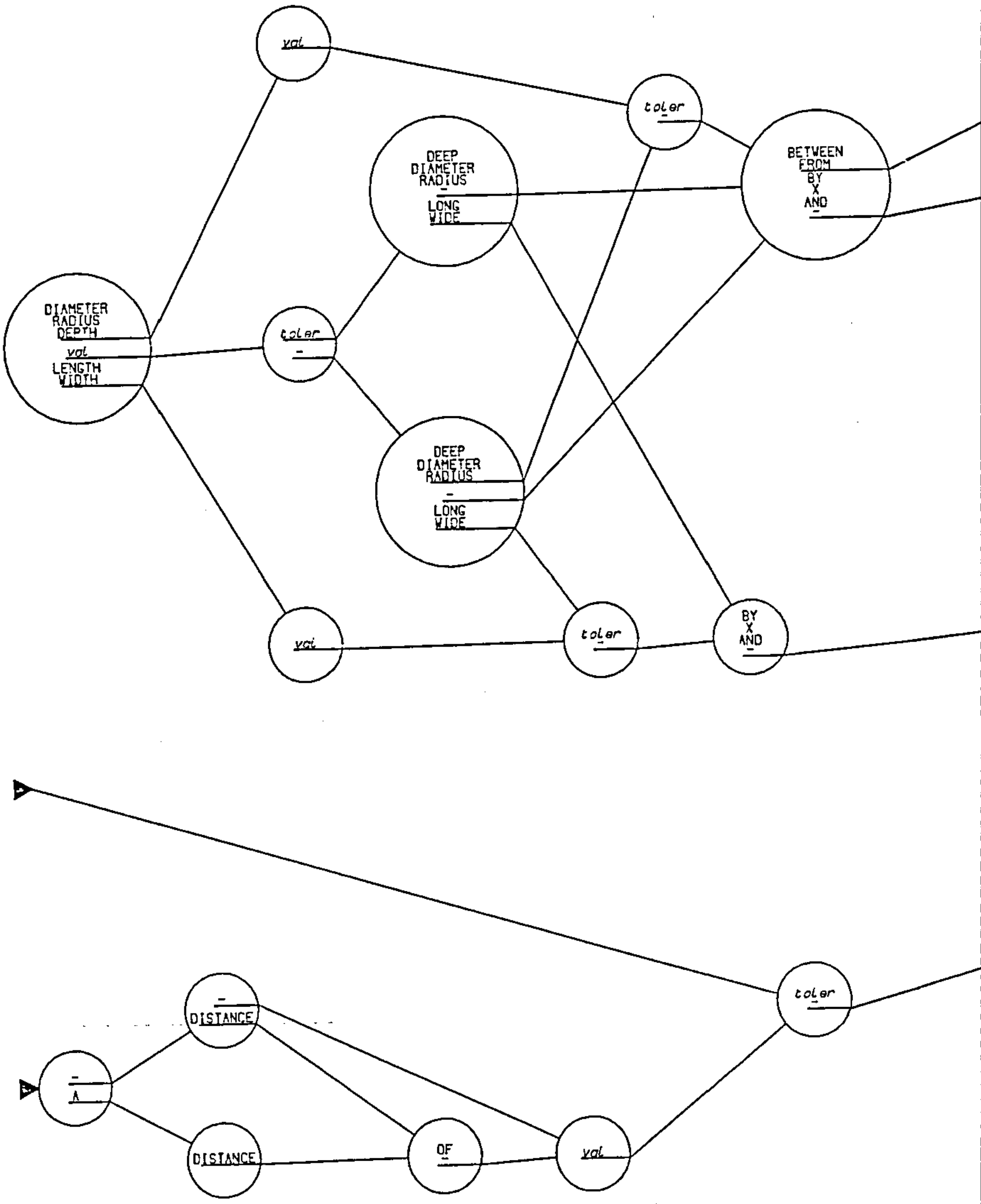
<u>step graph</u>



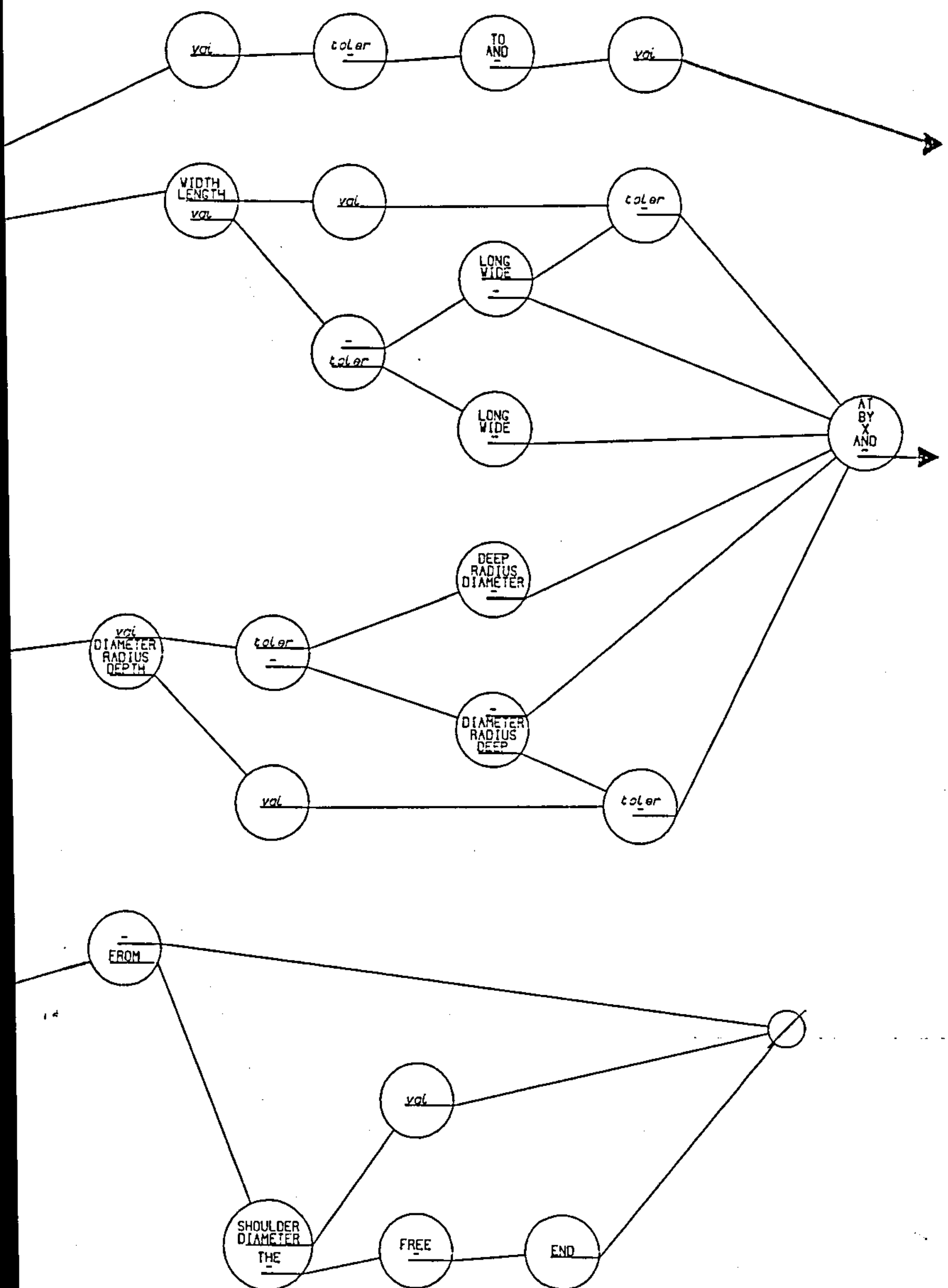
<u>reces graph</u>



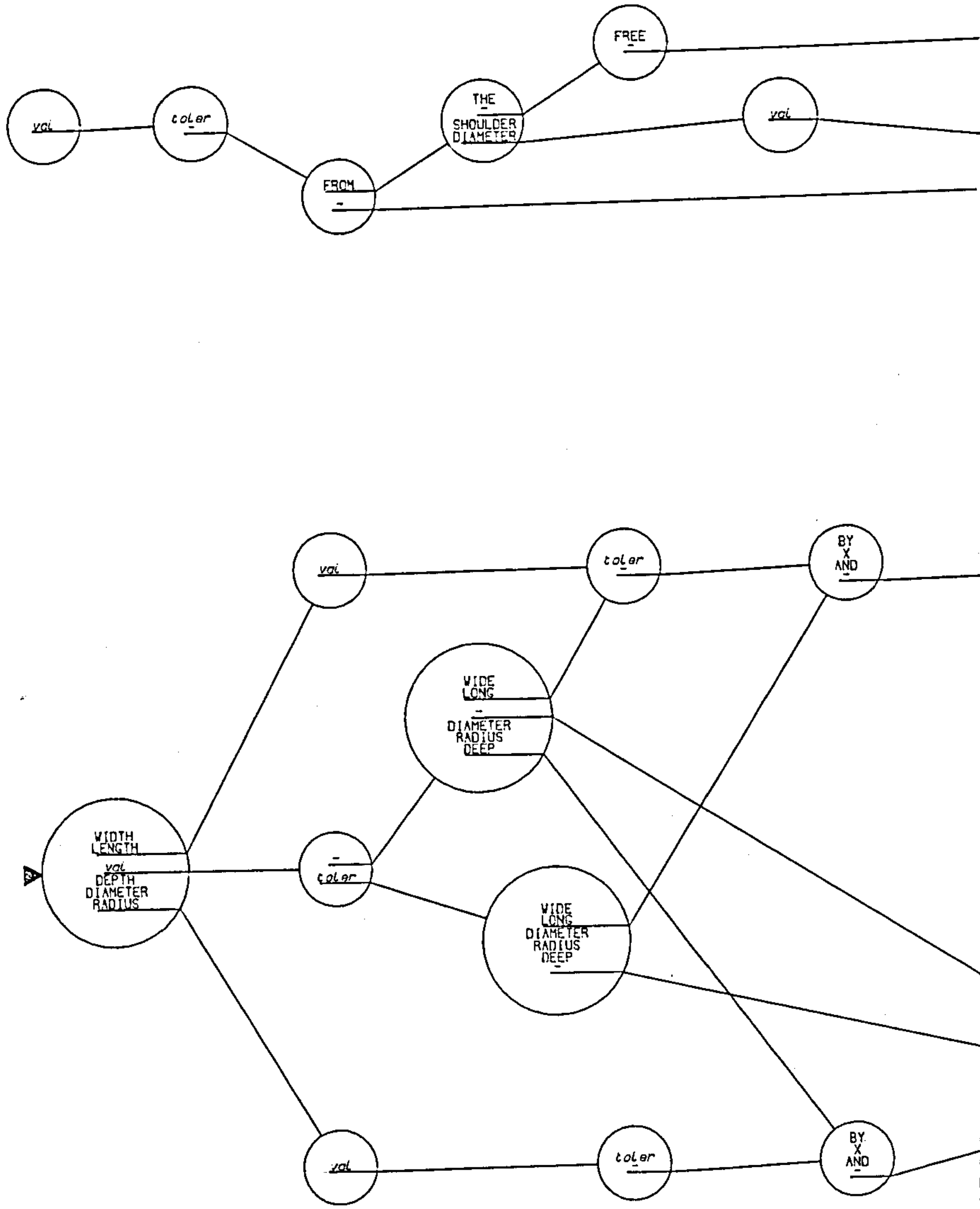
recsa subgraph



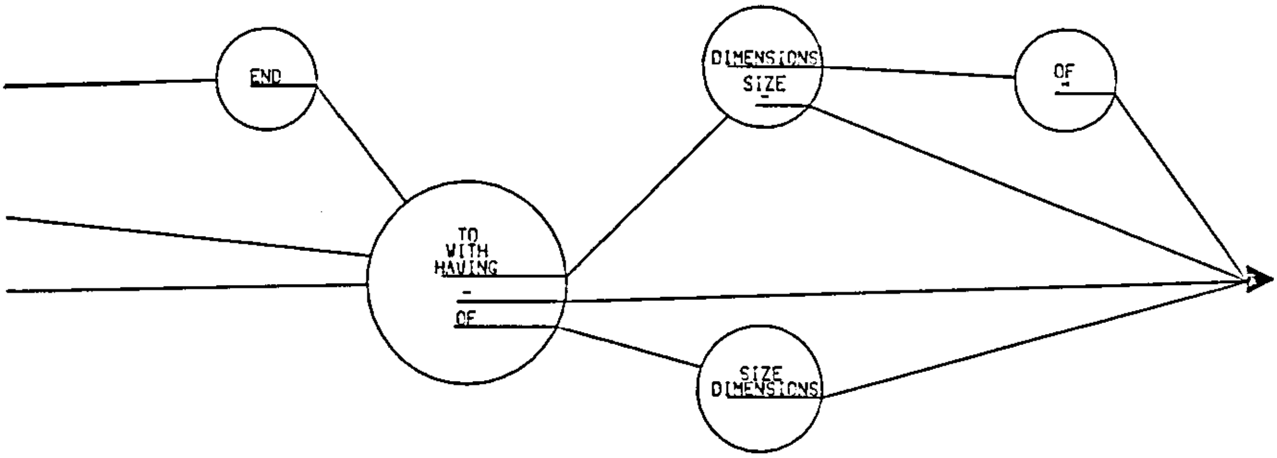
recsa subgraph

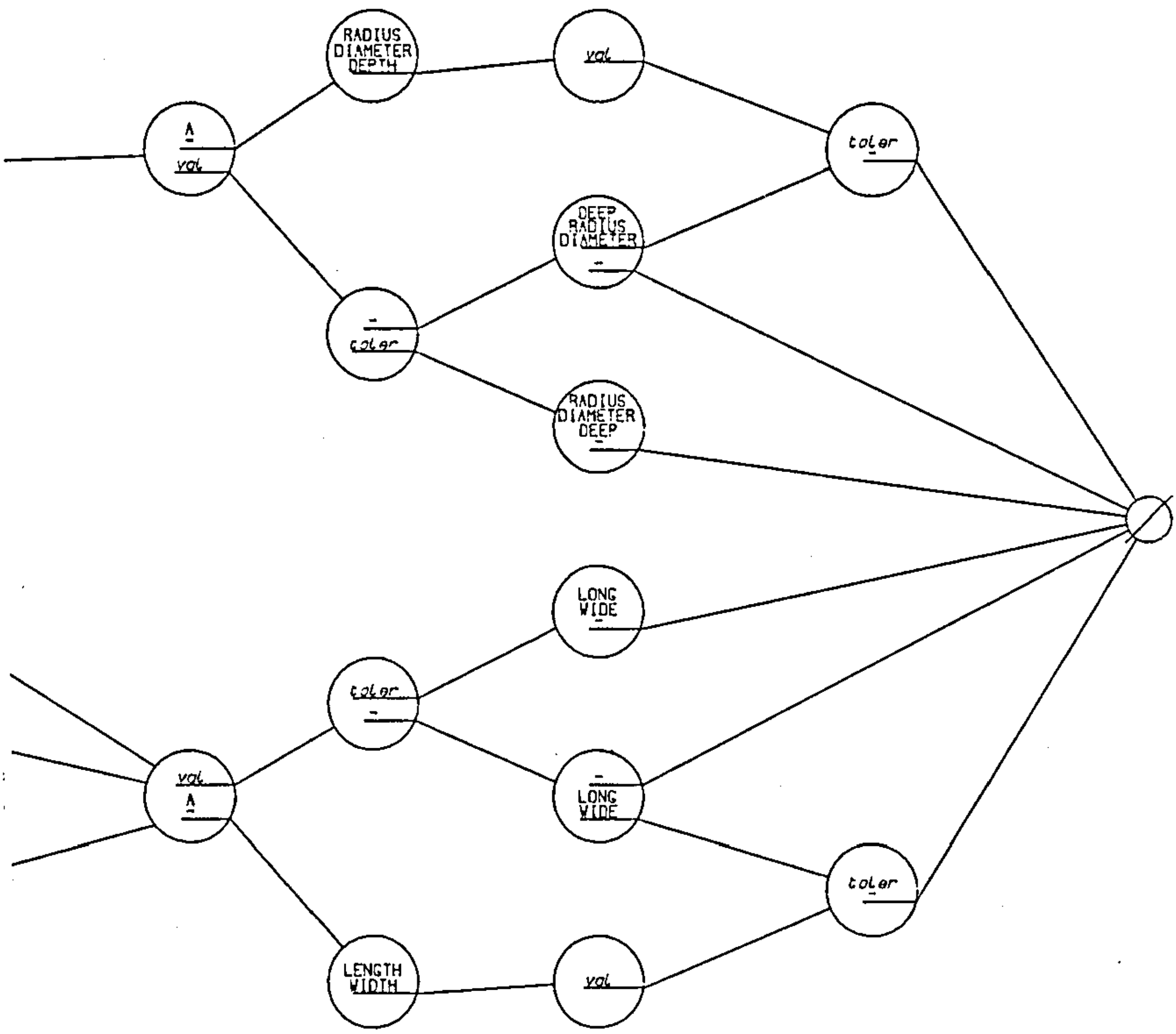


recsb subgraph

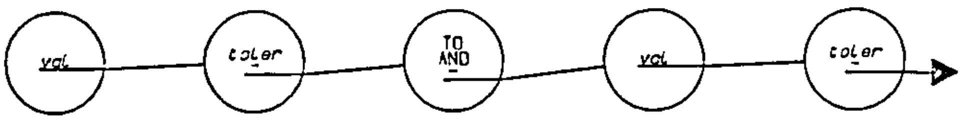


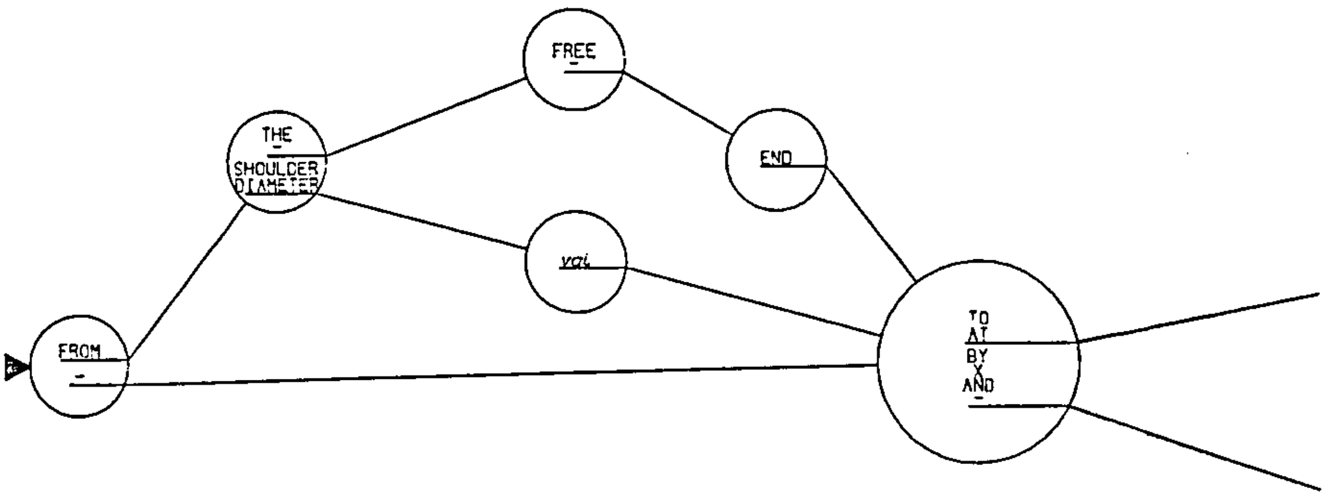
recsb subgraph



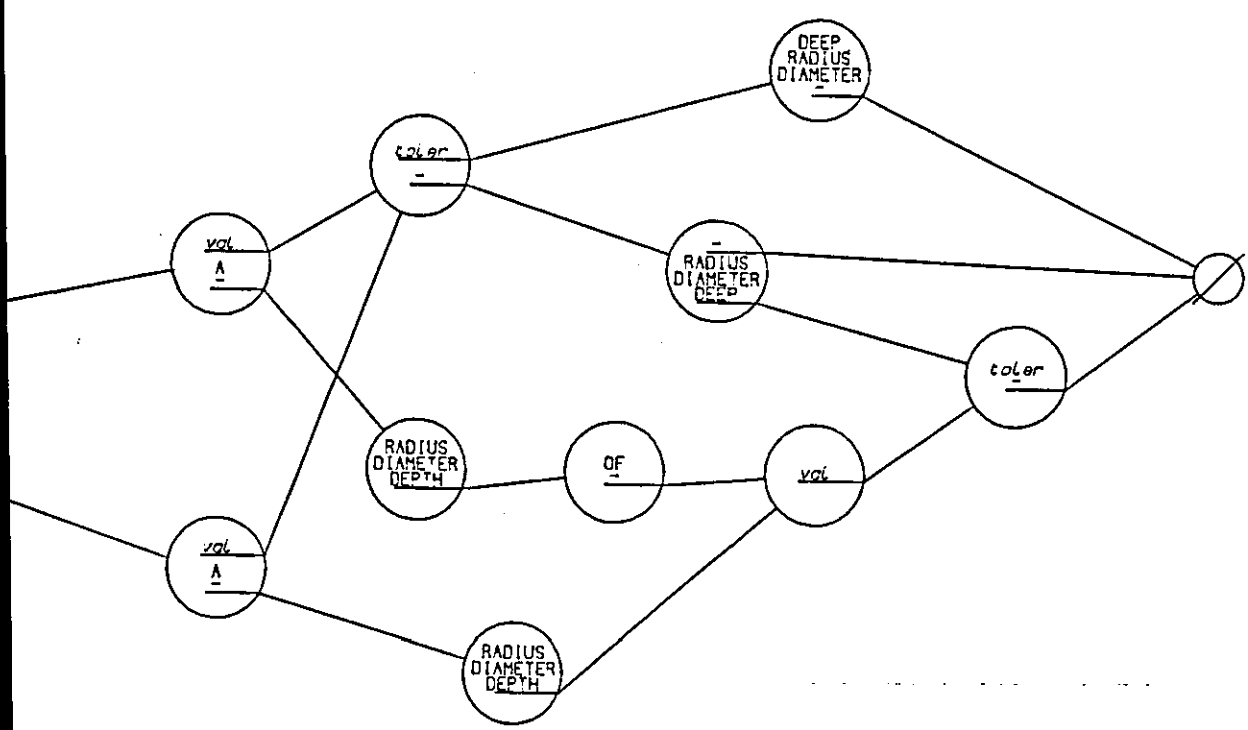


recsc subgraph

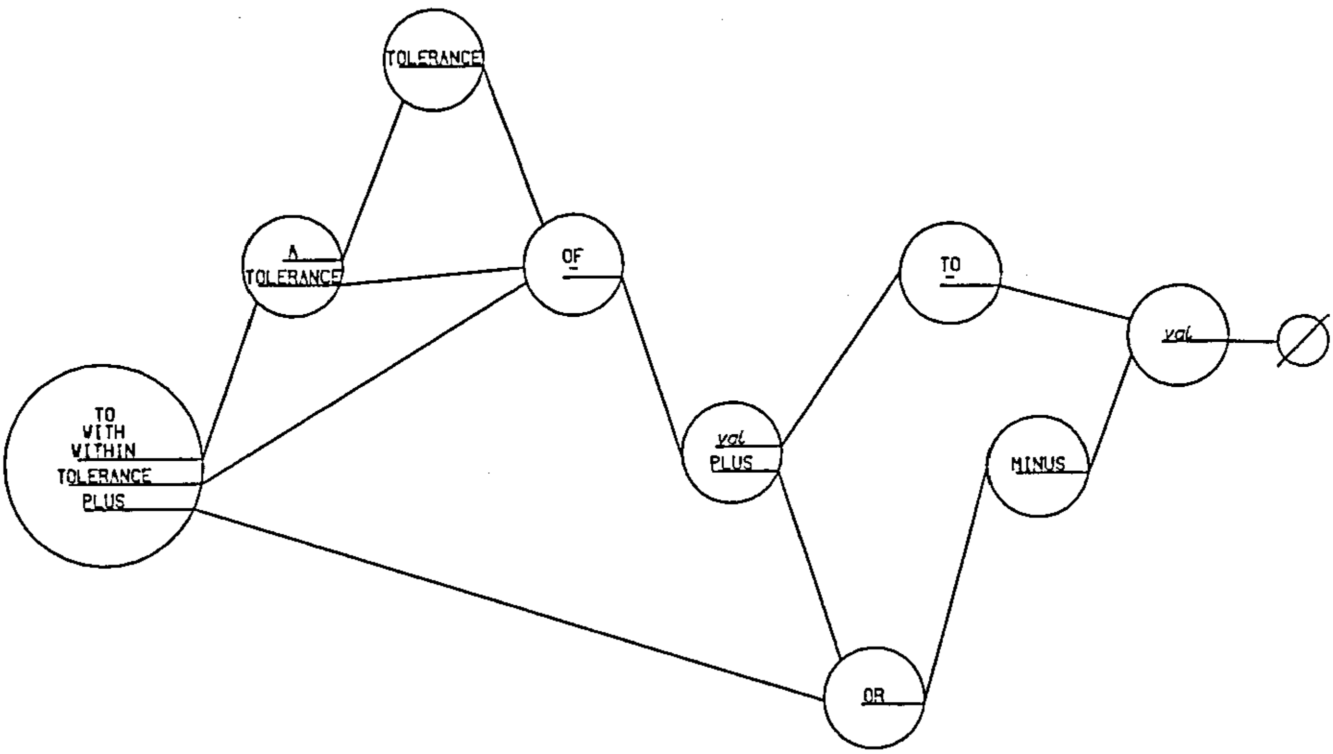




recsc subgraph



toler subgraph



5.3 From familiar engineering language to 3D solid model

Once the familiar engineering language graphs have been derived for an application, as in the turning example, they must be interfaced with a solid modeller in order to produce a design system. In the example of familiar engineering language for simple turning, the command processor GILT and the solid modeller BOXER have been used, although there is no reason why alternative techniques or software could not be employed. Naturally, the means by which other software would be interfaced might differ from the technique used to interface GILT and BOXER.

In order to interface GILT and BOXER it was neccessary to write a seperate application program (82) to extract information from the command processor and translate it into commands readable by the solid modeller. At the time of writing the application program, it was not possible to combine the command processor and the solid modeller into a single program. Therefore, an intermediate command file is used to store output from the application program for subsequent input into the solid modeller. The disadvantage of this approach being that truly interactive use of the example system is not possible. However, for the purpose of evaluating the feasibility of the technique, this is not a serious problem.

The use of GILT dictates to a certain extent how the application program to interface the two must be written, as conventions have to be observed if the command processor is to communicate with the application program. These include the choice of names and structure of subroutines, use of common blocks for communicating data and command processor initialization procedures. Obviously, if the solid modeller were to be combined into a single program along with the command processor and application program, then additional conventions, dependant on the solid modeller software, might also influence the design of the application program.

In the example of familiar engineering language for simple turning the procedure for interfacing the application program and the familiar language graphs can be summarized as follows:-

(a) Write language map in Graph Input Language

(b) Process through GILT to produce a blockdata file

(c) Process through CONVERT.UTY to convert blockdata file to standard FORTRAN77 format

(d) Compile and load application program, blockdata file, GILT libraries and FORTRAN77 libraries

(e) Run

The program CONVERT.UTY is necessary because the output blockdata statement produced by GILT is not of the correct format for use in a standard FORTRAN77 compiler. Therefore, the utility program CONVERT.UTY (82) has been written to convert the output blockdata file produced by GILT into the standard format.

5.4 Full integration with the solid modeller

Clearly a direct link between the command processor, application program and solid modeller would be more desirable and subsequent improvements in solid modeller software now make this feasible. For example, the recent availability of a subroutine version of BOXER enables the command processor, application program and solid modeller to be merged into a single program. Thus direct communication between the application program and the solid modeller is now feasible, making a truly interactive system possible.

The architecture of the subroutine driven solid modeller is slightly different to that of the command processor and does not require so many conventions to be observed in the application program. Information is communicated to and from the solid modeller data structure, not in common blocks, but in the argument lists of subroutine calls.

Although all the relevent geometric data for the solid model is held within the data structure of the solid modeller, at the time of writing the application program there was no way to interrogate it. Instead it was necessary to duplicate the information within the application program. Obviously this is an extremely inefficient solution, so the level of information duplicated in this way is kept to an absolute minimum, being limited to nominal dimensions only.

The recent availability of the subroutine version of the solid modeller makes it feasible to interrogate the geometric data of the solid model directly, thus eliminating the need for a duplicate data structure within the application program. This somewhat justifies the original decision not to waste time on this aspect of the program. However, even using the subroutine version of the solid modeller it is still not possible to incorporate tolerancing information into the solid model.

Tolerancing is of fundamental importance in engineering, for without information to describe the accuracy and to an extent surface characteristics of a component, it cannot be manufactured. Very recent work on Noname at Leeds University (81), implies that the facility to include tolerancing information into a solid model is soon to be available within BOXER (PAFEC Ltd is one of the companies sponsoring research on Noname). Presuming that a subroutine version will also be made available, it will shortly be feasible to fully integrate the tolerancing capacity in the language map with the solid modeller.

5.5 Geometric considerations

The concept and techniques underlying the familiar engineering language approach are consistent with the CSG input technique now supported by many solid modelling systems. The CSG approach is particularly appropriate to modelling engineering machining processes such as turning, enabling close simulation of actual metal removal operations. This is exploited in the application program, where the boolean difference operation is used to emulate metal removal from a predefined primitive solid representing the stock material.

The particular sub-set of geometric shapes encountered in the example of simple turning are simplified, requiring only simple cylindrical primitives to be modelled (see fig 5.11). A comprehensive turning system would have to include several additional features such as:-

(a) Non-cylindrical and previously worked stock material.

(b) External features like tapers, knurling and screw threads.

(c) Internal equivalents of the external features

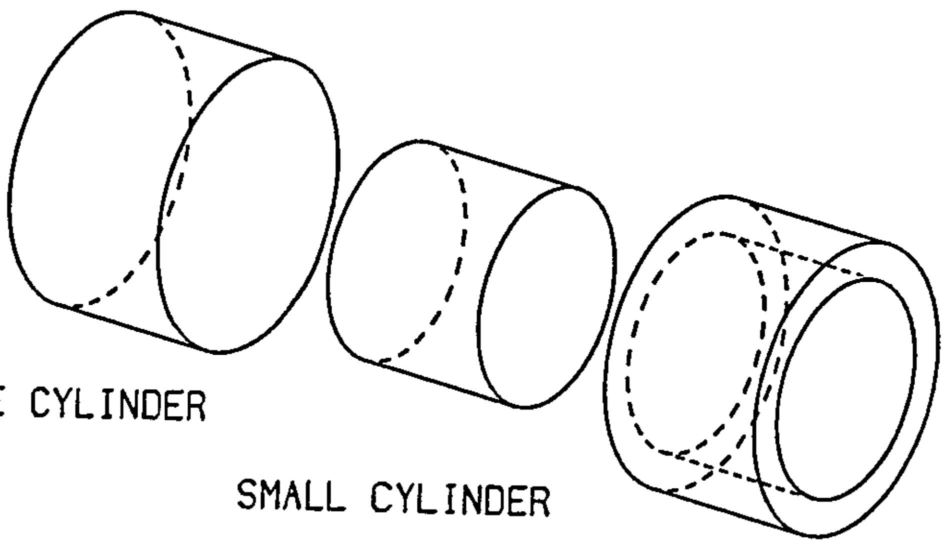


Fig 5.11 Modelling the removal of metal by simple turning

included in the simple turning example.

The definition of non-cylindrical stock material and the internal equivalents of external features presents few extra problems. The modelling of tapers is only marginally more difficult, as shown in fig 5.12. However, the representation of turned features such as knurling and screw threads is much more complicated.

The complexity of faithfully modelling features like screw threads and knurling is impractical with present generation solid modellers. Instead a symbolic representation must be employed similar to 2D engineering drawing practice. Consequently, the actual machining conditions encountered in these operations cannot be so closely emulated. To illustrate this point the geometric specification for a screw thread, shown in fig 5.6, can be modelled using conventional primitives as in fig 5.13.

5.6 Computer hardware snd software considerations

Many engineering components are specified by dimensions with a large number of digits which have to be stored and manipulated with a high degree of accuracy if a solid model is to be relied upon. The accuracy to which most computers work these days is sufficiently high, but in the case of the

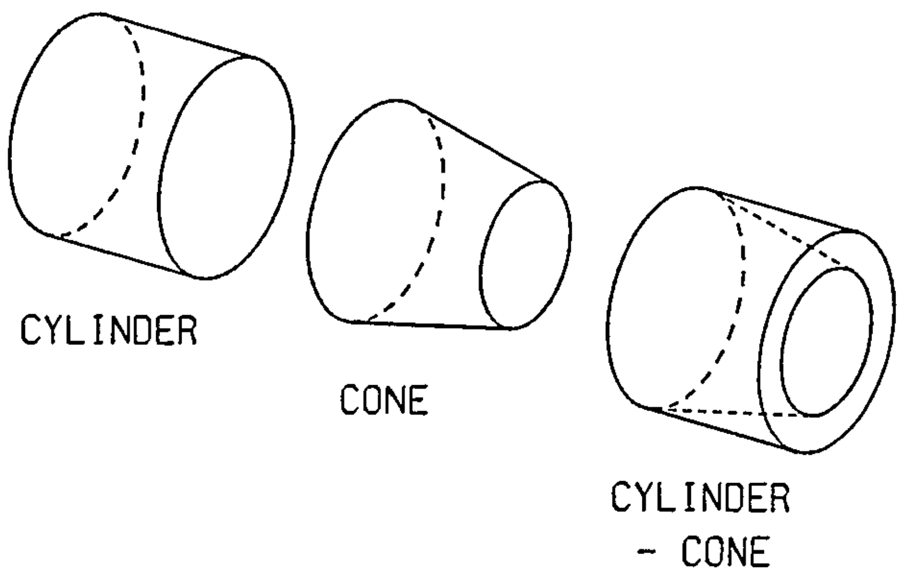


Fig 5.12 Modelling the removal of metal by taper turning

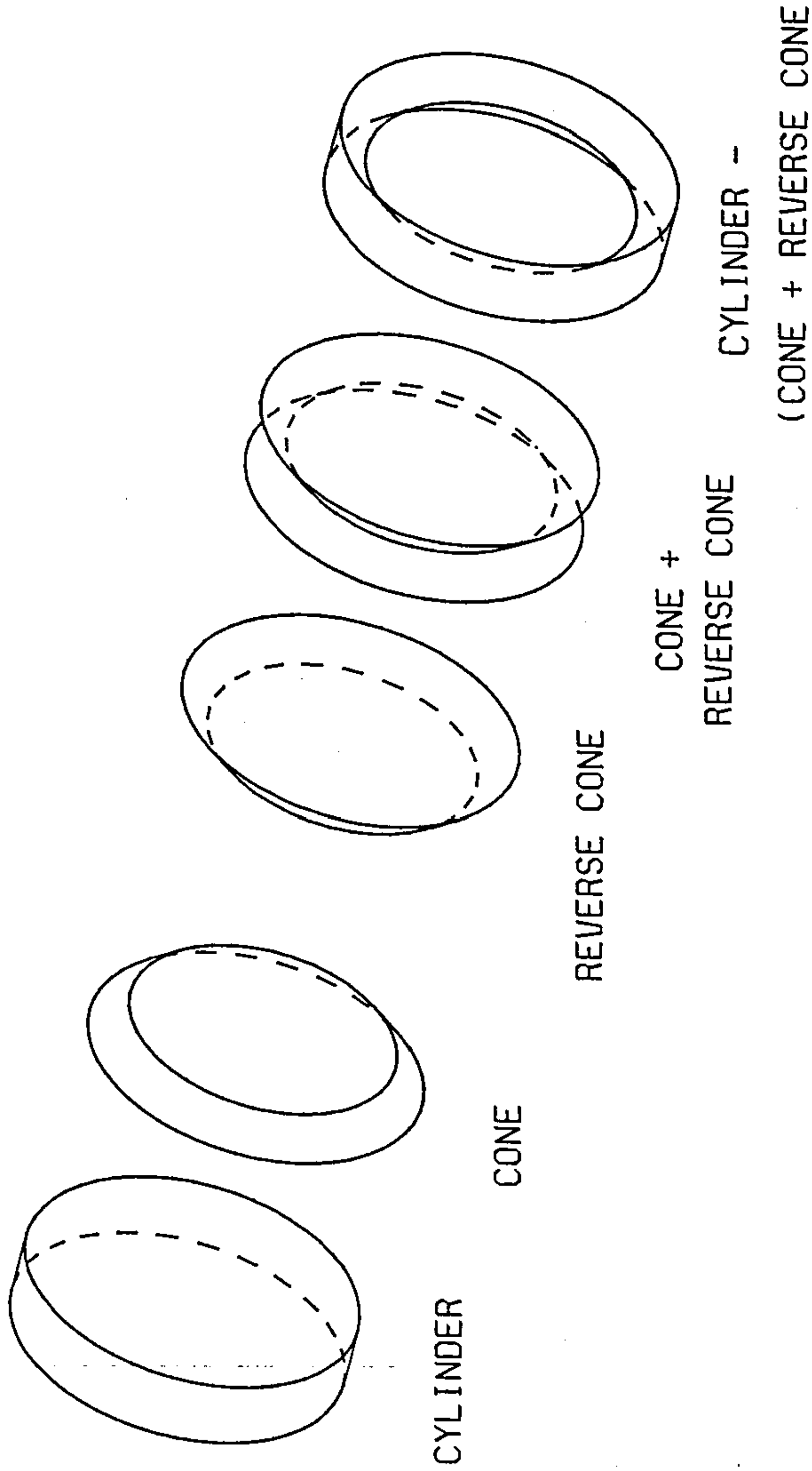


Fig 5.13 Modelling the removal of metal by screw turning

FORTRAN implementation on the university Prime computers the accuracy is limited to six digits only. Although it is possible to define double precision numbers with up to 12 reliable digits, the use of prewritten software such as GILT prevents this. As GILT has been designed for single precision numbers, the precision to which dimensions can be transfered to the application program and then to solid modeller is similarly limited.

Undoubtedly this shortcoming could be rectified by obtaining the source code for GILT and reconfiguring the variable declarations, but for the purpose of evaluating the familiar engineering language approach the problems of obtaining the source code, or commissioning changes to it, are not really justified.

Another significant limitation of the computer installation upon which the software is presently supported is the processing speed. The generation of many typical engineering components within the solid modeller results in impractically long delays, preventing true interactive use of the system. More powerful hardware would be a prerequsite for an industrial system, though given the rate at which available computational power is increasing this is likely to be a short term problem.

5.7 Observations on the nature of familiar engineering language graphs

Based on the experience of defining a relatively small sample of familiar engineering language graphs for turning operations it is not suggested that definitive conclusions can be drawn, however, certain fundamental relationships are evident. These relationships are equally relevant to other processes in which a familiar engineering language might be applied.

5.7.1 The four basic elements of most familiar engineering language commands

OPERATION + LOCATION + DIMENSIONS + TOLERANCES

Apart from the operation which is always specified first, the sequencing of the elements can vary widely, locations can be implicit and tolerances left to default, but unless the four main elements can be identified there is no operation.

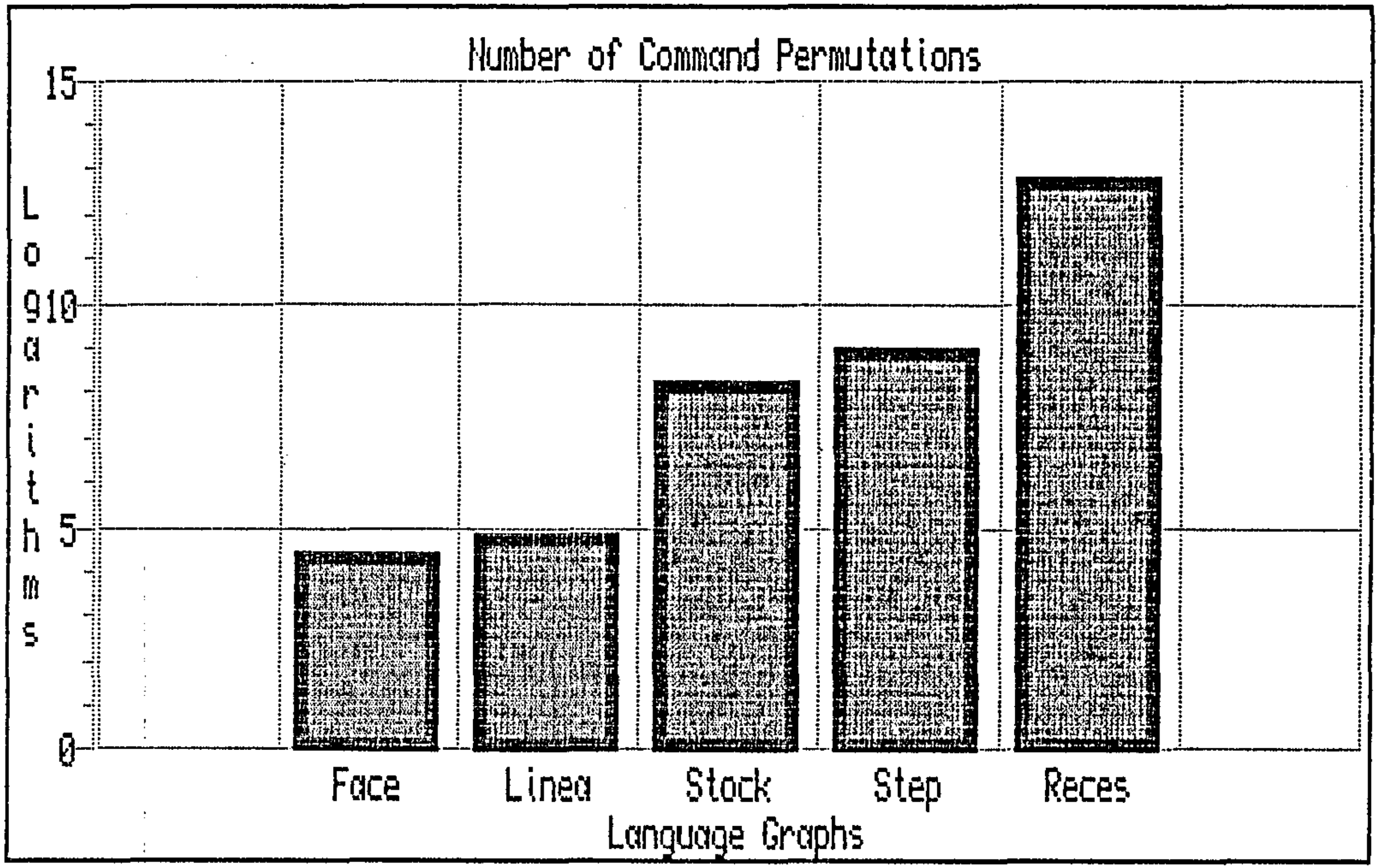
5.7.2 Repeating patterns in the familiar language graphs

Many of the example graphs exhibit similar sub-constructions, that is patterns of states that are consistently repeated. Although, it is not usually feasible due to subtle variations in terminology or potential loss of overall clarity to create a subgraph. Probably the best example of this can be seen in comparing the lower halves of 'stock', 'step' and 'recsb' (see graphs 5.2, 5.3 and 5.8). Nevertheless, recognising repeated patterns speeds up the process of deriving new graphs.

5.7.3 Order of complexity is proportional to number of dimensions

The number of dimensions that must be specified in relation to an individual operation would appear to have dominant influence on the size and complexity of the resulting familiar language graph. If a comparison is made between the number of command permutations generated by a graph and the number of dimensions that need be specified in order to define the relevent operation, then a trend is indicated.

With reference to fig 5.11, note that graphs 'face' and 'linea' require only one dimension to specify the operation, graphs 'stock' and 'step' require two dimensions, whereas 'reces' requires three dimensions. Associated with each order of dimensions, is a corresponding increase in the number of command permutations, which is approximately a magnitude of ten to the power four. Therefore the idea is proposed that there is a crucial relationship between the number of dimensions necessary to specify an operation and the number of command permutations in the corresponding



Graph 5.11 Order of complexity for different graphs

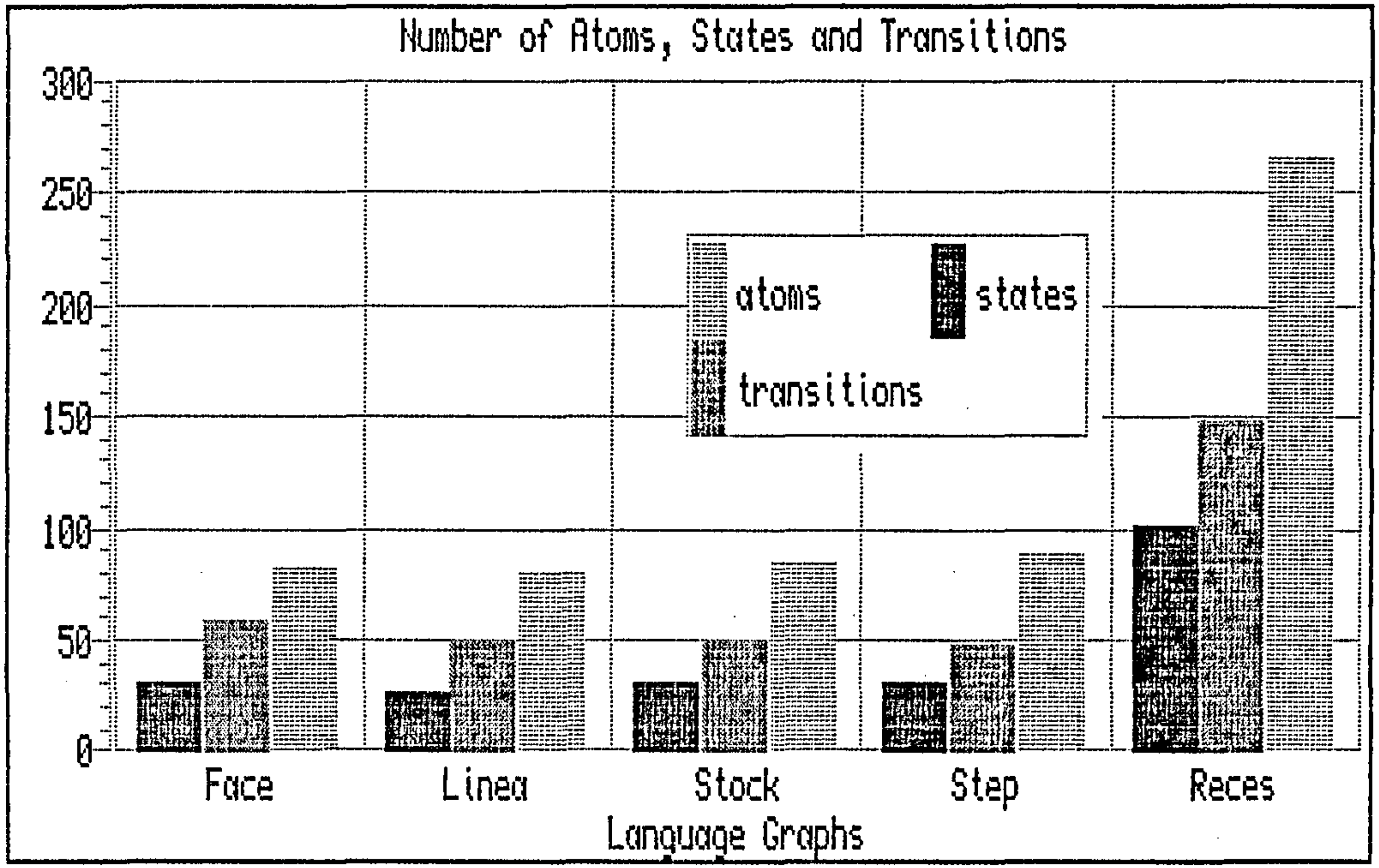
familiar language graph.

Examining the construction of the language graphs themselves, does not provide any clear evidence of the order of complexity, see fig 5.12. The number of states, transitions or atoms in a familiar language graph is not a very good indication of it's complexity. For example, the graph face appears complex because there are a large number of states and transitions, although the facing operation is only a first order graph.

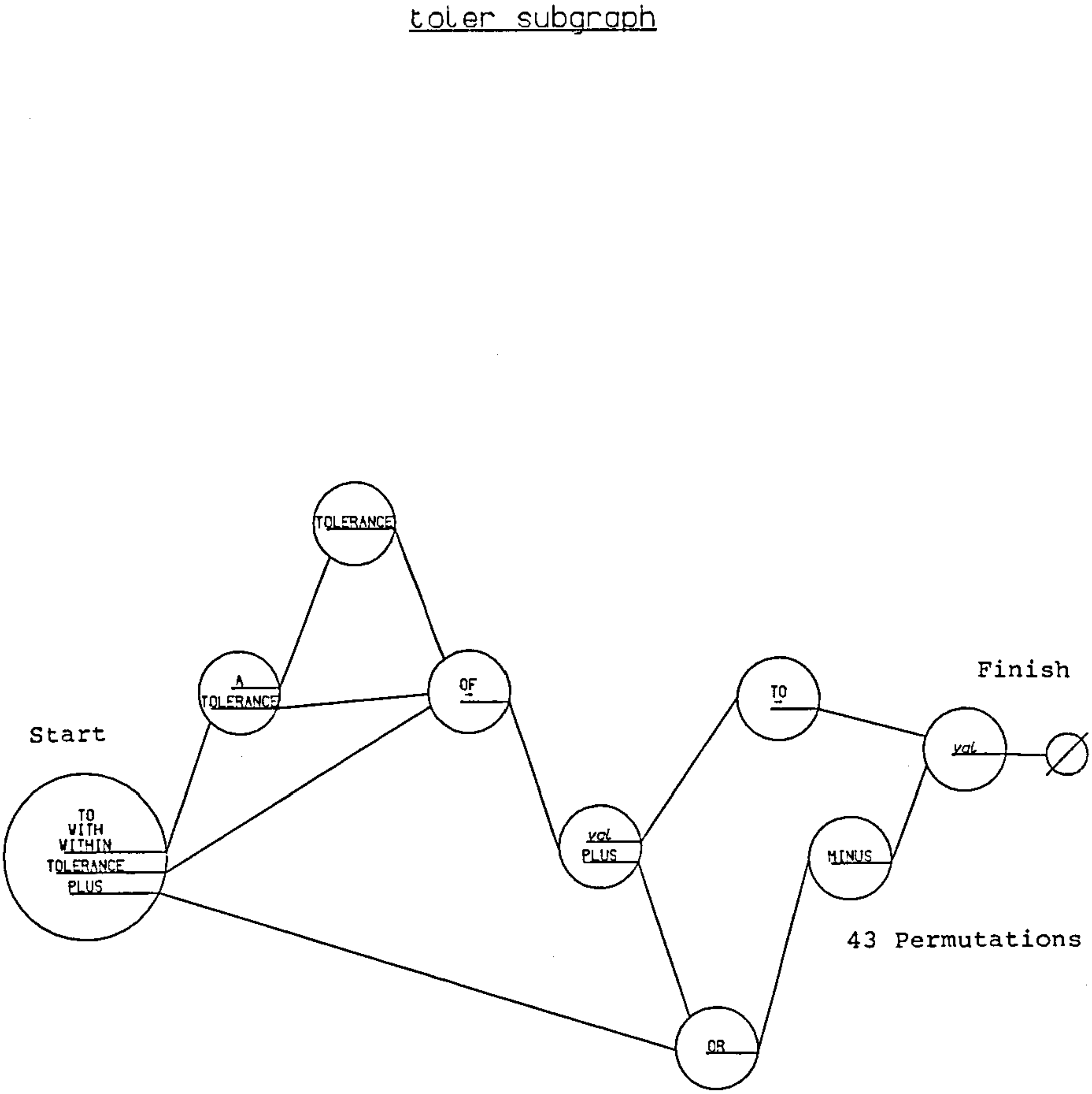
It may be argued that the evidence of figs 5.11 and 5.12 merely indicates that a high level of consistency was achieved in deriving the familiar language graphs. On the basis of such a small sample it is not possible to be certain that this is not the case.

5.8 The technique used to calculate the number of command permutations in a familiar language graph

In calculating the number of command permutations that are generated by each graph, a simple technique is used that is best explained with reference to an example. The subgraph for tolerancing has already been described in detail in the previous chapter, therefore, the procedure for calculating the number of command permutations is shown in fig 5.14.



Graph 5.12 The number of atoms, states and transitions for different graphs



(I) Working from start highlight all transitions for more than atom - in { }'s

(II) Calculate number of routes to each state, progressively through graph using:
multiplication for states in series
addition for states in parallel

(III) On large graphs it is helpful to split them into sections - identify key states where transitions converge

Fig 5.14 Calculating the number of command permutations

Having identified the starting and finishing points of the language graph, depending on the complexity of the graph, it is advantageous to identify intermediate key states, where several transitions converge (state four on the toler graph is a potential key state). Then between two key states in turn, each transition must be examined to determine the number of atoms associated with it. Where more than one atom can be identified it helps to note the number at the midpoint of the transition.

Using the rule that serial transitions are multiplicative and parallel transitions additive, the number of permutations between two key states is easily calculated. Again it helps to note these sub-totals at the terminating key state. Then applying the same multiplicative/additive rules to these subtotals the total number of permutations for the whole graph can be determined.

Notice that while graphs like 'stock' are self-contained, other graphs like 'reces' are not and to make an accurate computation of the number of permutations for these, examination of the 'refer' graph is also necessary. Remembering also to take into account the number of combinations in any subgraph which is called en route.

5.9 Graph drawing and expansion utilities

Before leaving this chapter, mention must be made of two software tools developed to assist in the drawing and expansion of familiar engineering language graphs.

A program written to drive a 2D CAD system (82), enables finished familiar engineering language graphs to be drawn from Graph Input Language semi-automatically. Taking the manual development draft of a familiar engieering language graph, and determining the coordinates of the centres of each state (bubble) for input into the drawing program, a finished drawing of the langauge graph is automatically produced.

Although the responsibility for roughly organizing the relative positioning of states rests with the user, this is an inevitable consequence of graph development and optimization anyway. Therefore, the use of the familiar language graph drawing program dramatically reduces the time required to produce finished quality drawings.

Another program has been written (82), which enables the Graph Input Language for a given familiar language graph to be expanded for all possible command permutations. For all practical purposes, graphs beyond first order complexity are too large to expand completely, although it is possible to isolate sub-constructions of higher order graphs for expansion. The output of graph expansions is an aid in the development of familiar language graphs, and also provided the means to verify the technique used in manually calculating graph complexity for low order graphs.

Finally, a program to simplify the creation of Graph Input Language files has been developed (82), which also presents them in a suitable format to be read by the utility programs.

6 QUESTIONNAIRE AND RESULTS

In order to obtain feedback and evaluate the ideas expounded in this thesis, the opinions of experienced engineers with a wide range of skills were sought. In the circumstances, the only feasible means of accomplishing this was to prepare and administer a questionnaire.

Many factors had to be taken into account in planning the questionnaire, but it was considered important that it should:-

(a) be sufficiently concise to ensure the goodwill of participating companies.

(b) present questions in a practical rather than an academic context.

(c) be appropriate to a wide variety of disciplines (shop-floor to design office).

(d) allow a high accuracy of completion.

(e) give an insight into the utility of the familiar engineering language approach.

With these in mind, two experimental questionnaires were drafted (see appendices A and B) and tested using a group of sixteen final year engineering students. The eventual format of the questionnaire was decided, based on the experience gained in administering these questionnaires.

The first of the two experimental questionnaires was designed to simulate the use of familiar engineering language, by asking respondants to annotate a series of drawings detailing the manufacture of a simple turned component. However, the questionnaire proved difficult to administer and the results hard to assess because of the potential for ambiguity using this kind of approach. The results also varied enormously in quality and quantity, indicating that the questionnaire was neither easy or pleasant to complete.

The second of the experimental questionnaires was much more successful than the first. It was designed to test the output from a familiar engineering language driven design system, in the form of a process plan. Using two 'real' process plans for simple turned components as benchmarks, a third was synthesized using the familiar engineering language driven system described in chapter six. In choosing the benchmark process plans, permission was granted to visit two companies, randomly selected from those known to the university, to obtain a 'typical' process plan for simple turned components.

The results obtained from the second experimental questionnaire, were consistent and quantifiable although certain shortcomings were still evident. A proportion of the questionnaires were not completed correctly and thus void. Certain errors and potential ambiguities were also highlighted.

To reduce the number of void questionnaires and generally encourage a higher quality response, it was necessary to administer the questionnaires individually. If not statistically ideal, for practical reasons it was not feasible to interview more than fifty engineers in this way. Of the fifty, it was planned to sample an equal quota of designers, production engineers, technicians, machinist/ setters and researchers. Within, and between, occupations, the age and experience of respondants was to be randomly distributed.

The questionnaire was computerized to ease correlation of the results, with the additional and significant advantage that questions could only be answered in the planned sequence. That is to say it was not possible to prejudge the outcome or modify answers in light of subsequent questions. The use of computerized questionnaires have been found to have other advantages too (80), for example, they are more enjoyable to complete.

In translating the questionnaire from a purely academic enviroment to a practical one, it was necessary to make minor alterations to the synthesized process plan. Most notably, this included the provison of a parting off operation not explicitly incorporated into the example of familiar engineering language for simple turning. A general tolerance of plus or minus Ø.1Ømm was also presumed for completeness.

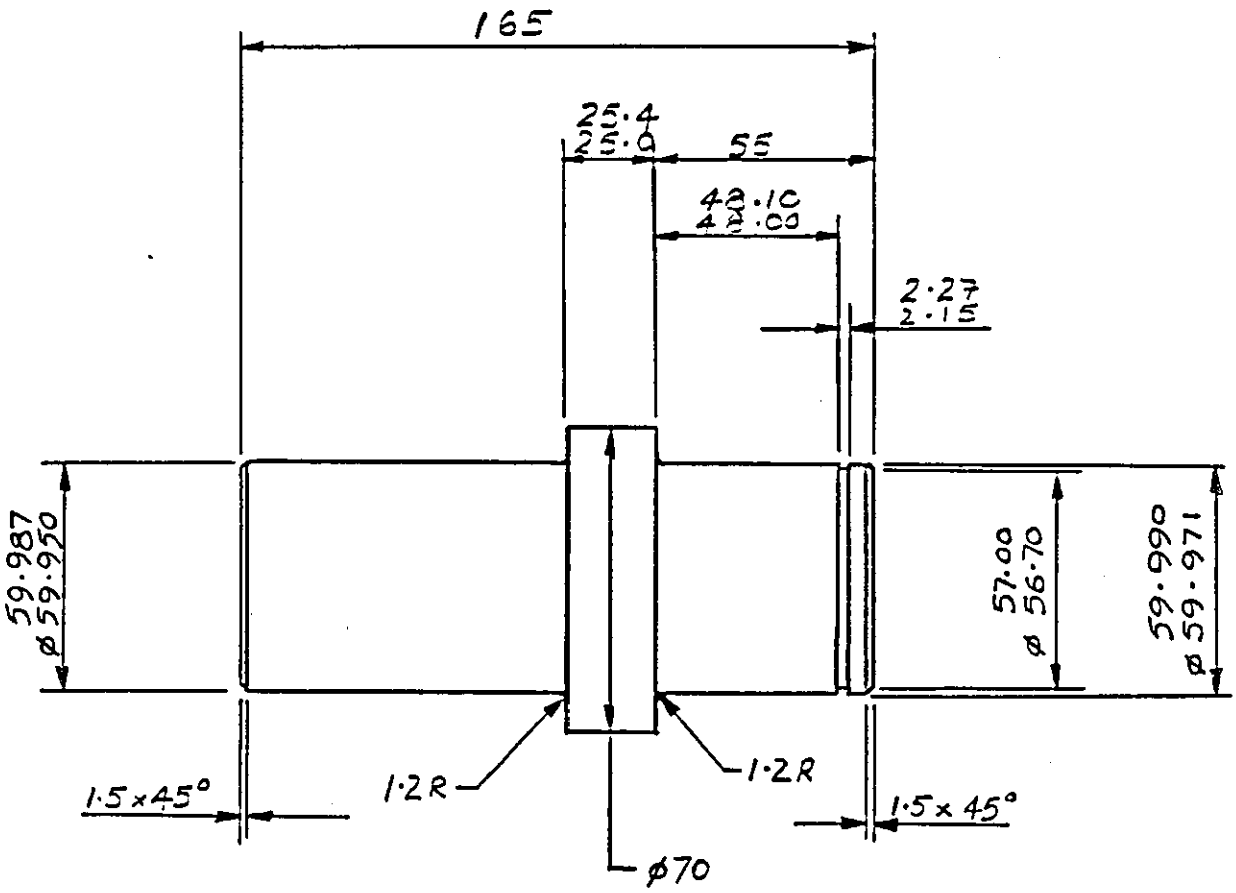
6.1 Presentation of the results

The questionnaire and results, in graphical format, are presented on the following pages, with the complete set of answers included in appendix C.

INSTRUCTIONS

<u>Compare three different styles of process planning</u>

On the following pages are extracts from three different process plans for simple turned components. After a careful study of each of the process plans, please continue the computer questionnaire, giving answers which reflect the impressions you have formed of them.



PROCESS MASTER

Description

12.5T Runner Pin

Operation 1

Face Centre Turn 59.990/59.971 Leave Grinding Allowance .25

Chamfer Groove 2.27/2.15 Wide X 57.00/56.70 Diameter Part Off

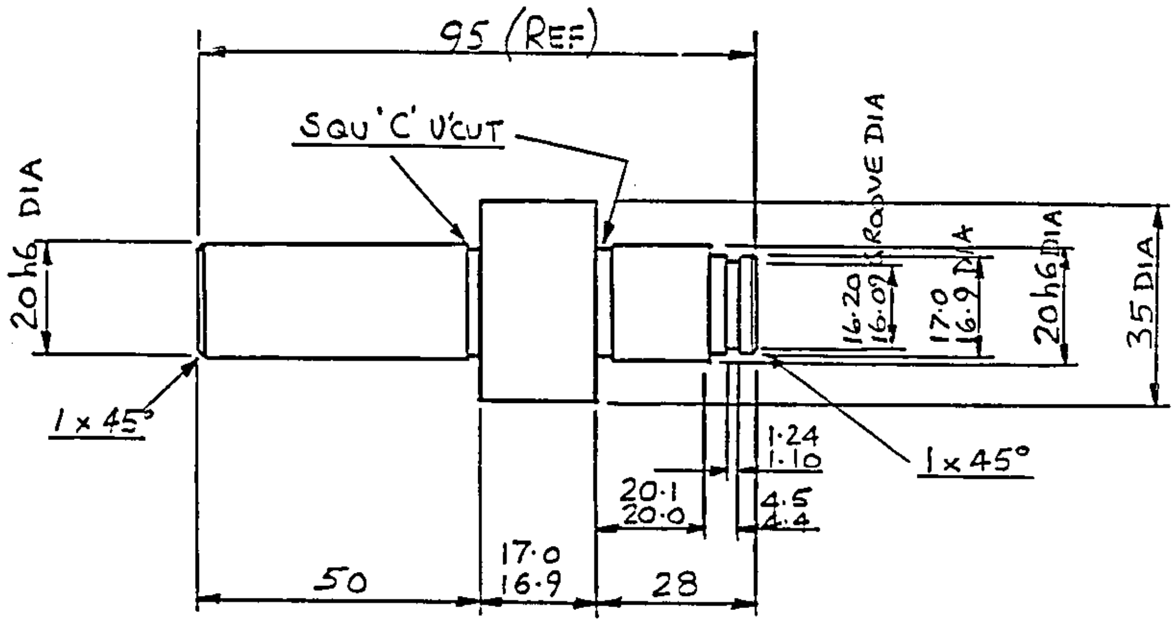
to 165 lg.

Operation 2

Hold in collets Turn 59.987/59.950 Leave Grinding Allowance

.25 face Chamfer & Centre

Fig 2.13 A typical format for a process plan: Davy Morris



Job Specification

Description

BRT MS ROD BS97Ø 22Ø MO7 35 DIA TOL SMALL H1Ø

Operation Description

Turn face U/cut Groove Chamfer and Part off CPTE

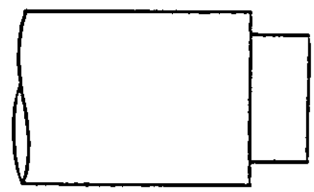
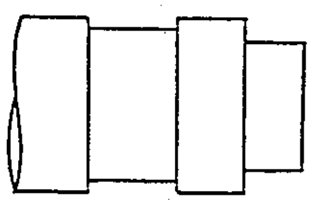
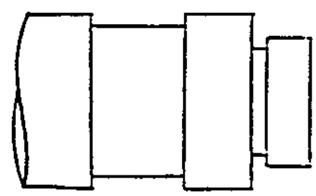
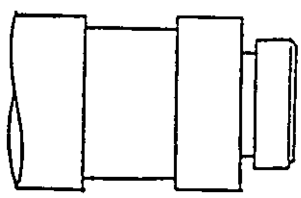
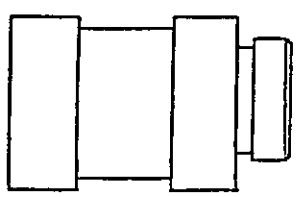
Remove sharp corners

C

Process Plan

General Tolerance: +/- 0.10mm

Stock material mild steel round bar diameter 63.5 x 1000 long

1 	2 
Turn the outside diameter of the bar down to 60	Turn a step in the bar having diameter 40 tolerance 40.00 to 39.95 and length 20.00 Plus or minus 0.05
3 	4 
Turn a recess in diameter 2 with dimensions of width 32 and diameter 52 at a distance of 24 from diameter 1	Turn an undercut at a distance of 16 from the end having width 4 and depth 4
5 	6 
Chamfer the end of the bar width 2 at an angle of 45^{o}	Part-off to length 100

(1) Please indicate your age group

1) up to 19 2) 2Ø-29 3) 3Ø-39

4) 4Ø-49 5) 5Ø-59 6) 6Ø plus

(2) What is your occupation?

1) Designer 2) Machinist/setter

3) Production engineer 4) Technician

5) Researcher 6) Student

(3) How many years experience in engineering do you have?

1) less than 1 2) 1 to 2 3) 3 to 5

4) 6 to 1Ø 5) 11 to 2Ø 6) 21 plus

(4) During design, how important do you think it is to consider how components will be manufactured? Indicate on the following scale:-

Ø 5 1Ø

irrelevant | . . . . | . . . . | essential

(5) How much influence do you think process plans have on the way components are manufactured? Indicate on the following scale:-

Ø 5 1Ø

none | . . . . | . . . . | total

(6) ** NOW READ INSTRUCTIONS ** How competent are you to compare the process plans? Mark your ability on the following scale:-

```
           Ø         5         1Ø
novice     | . . . . | . . . . |  expert
```

(7) In manufacturing terms how similar do you think the three components are? Indicate on the following scale:-

```
           Ø         5         1Ø
opposite   | . . . . | . . . . |  identical
```

(8) Rank the process plans according to how helpful you find them:-

1...............

2...............

3...............

(9) Rank the process plans according to how helpful you would find them if the text was removed (ie drawings only):-

1...............

2...............

3...............

(1Ø) Rank the process plans according to how helpful you would find them if the drawings were removed (ie text only):-

1...............

2...............

3...............

(11) How familiar are you with the wording used in process plan C? Indicate on the following scale:-

	Ø		5		1Ø	
unknown	|	. . . . .	|	. . . . .	|	completely

(12) How important do you think it is to use a traditional style for process plans? Indicate on the following scale:-

	Ø		5		1Ø	
irrelevant	|	. . . . .	|	. . . . .	|	vital

In total fifty-six questionnaires were administered, of which six were found to be void for various reasons. For a long time it looked as though there were fifty-one correct questionnaires, until it transpired that one respondent had indicated that he was between twenty and twenty-nine years of age, with twenty-one plus years of experience! This reduced the number to fifty which somewhat fortuitously equalled the initial target.

Obtaining the cooperation of engineering companies to administer the questionnaires was not easy, and therefore it was not possible to control the attributes of the respondents as closely as originally intended. For example, more production engineers and technicians were sampled, approximately sixty percent, than designers or machinists. Similarly, seventy-two percent of respondents had more than ten years engineering experience, with thirty-six percent having in excess of twenty.

To offset these problems, in certain instances age and experience groups were rationalized and this should be evident from the graphs. In many cases sample sizes were unavoidably different and therefore a comparison of the average answers for each group has been made. Naturally the average of a group containing six members, like designers, is less precise than a group containing sixteen (eg production engineers), so this should be born in mind.

Analysis of the results can be broken down into three principle sections:-

(a) Opinions

(b) Preferences

(c) Comments

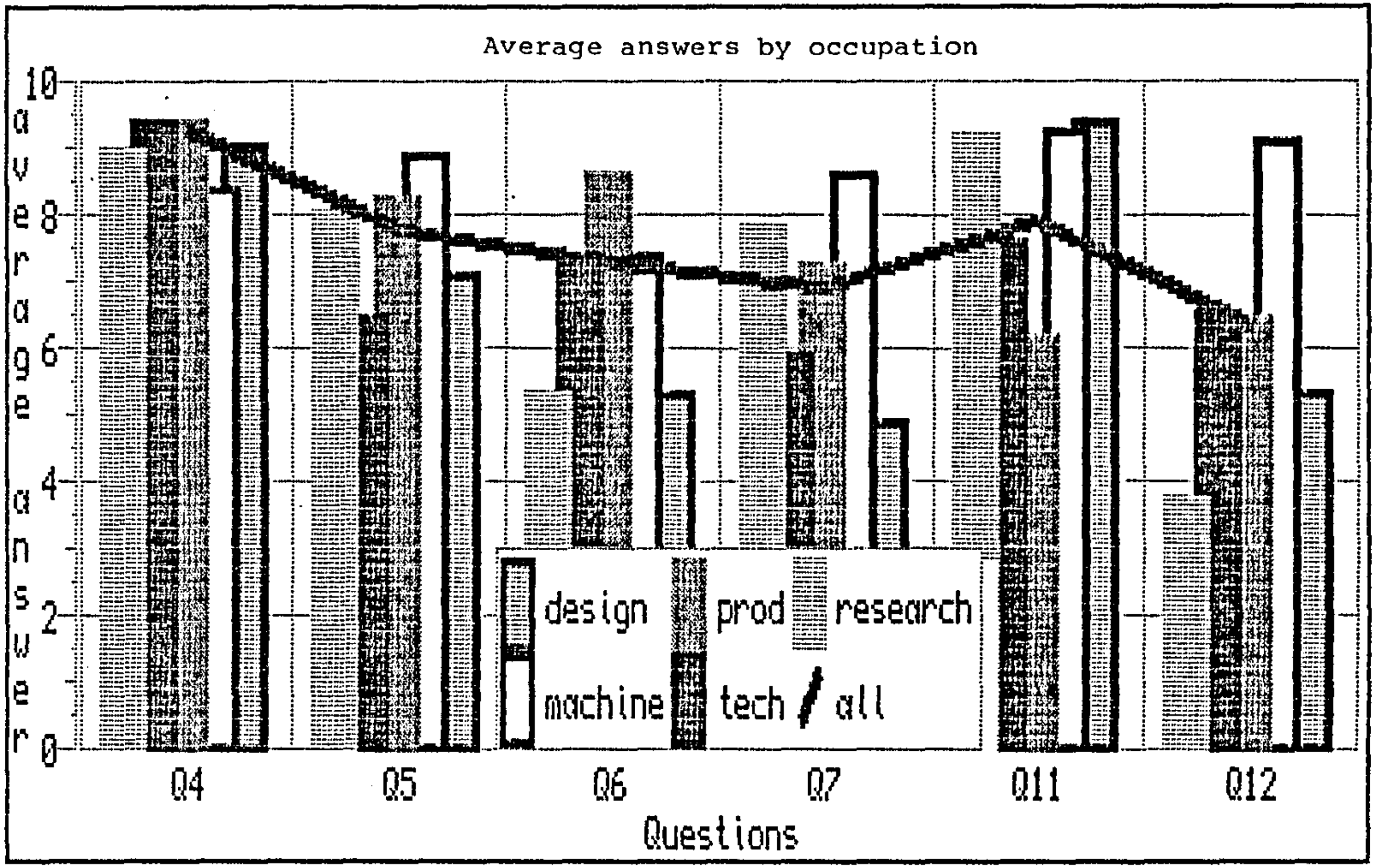
6.2 Analysis of opinions

'Opinions' refer to those questions which asked the respondent to indicate the strength of his opinion in relation to two extremes on a linear scale of one to ten. This technique was employed in questions four, five, six, seven, eleven and twelve (see questionnaire).

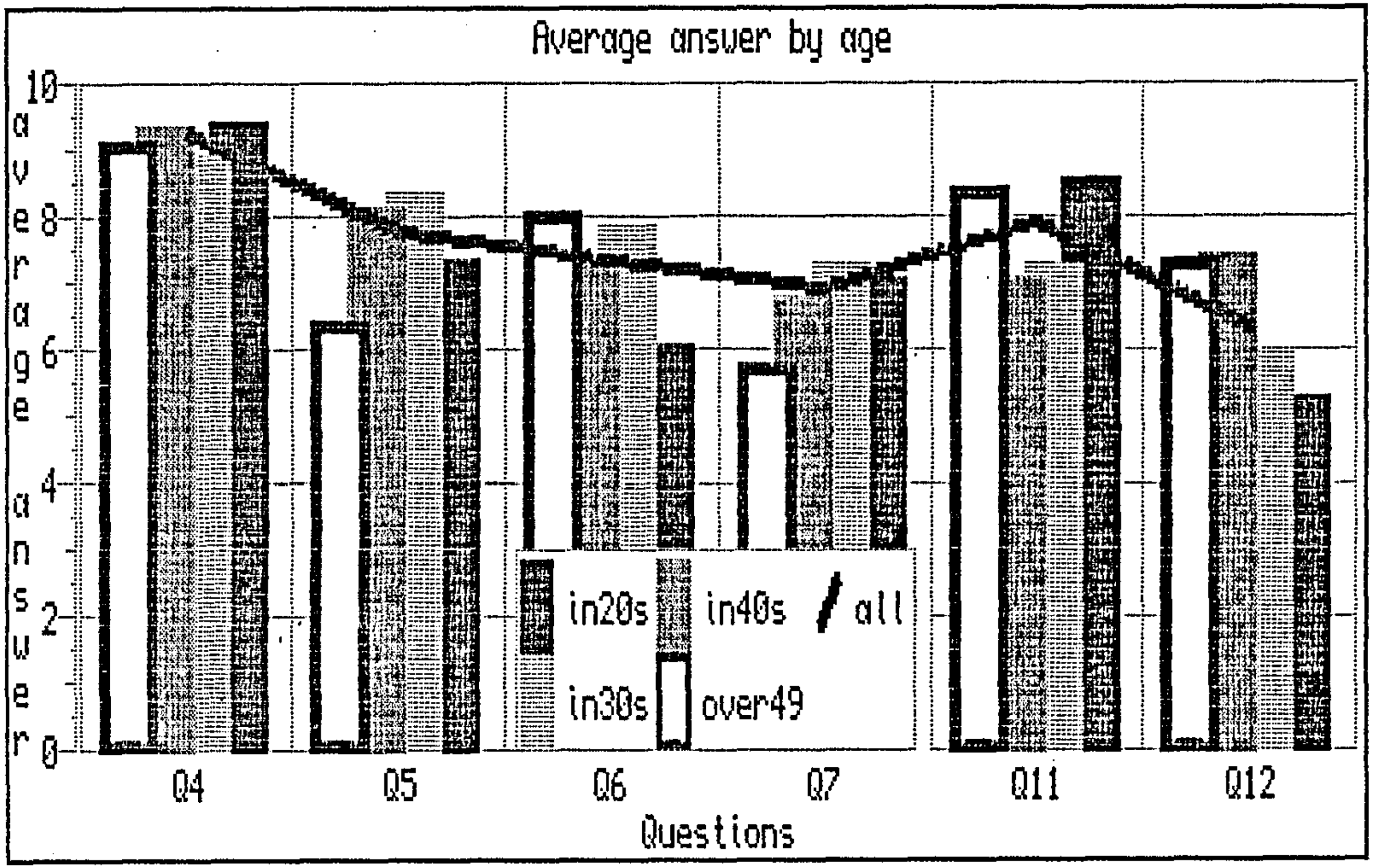
From the results, the average answer of all respondents has been derived for each of the six questions, and is superimposed as a line on the graphs 6.1-3. These average answers are then broken down into the average answers for each occupation, age group and experience group. Notice that the age groups and experience groups have been rationalized because of the reasons outlined above.

Looking at the graphs, clearly there is wide and consistent agreement, whether analysed by occupation, age or experience, that it is virtually essential to consider how a component is

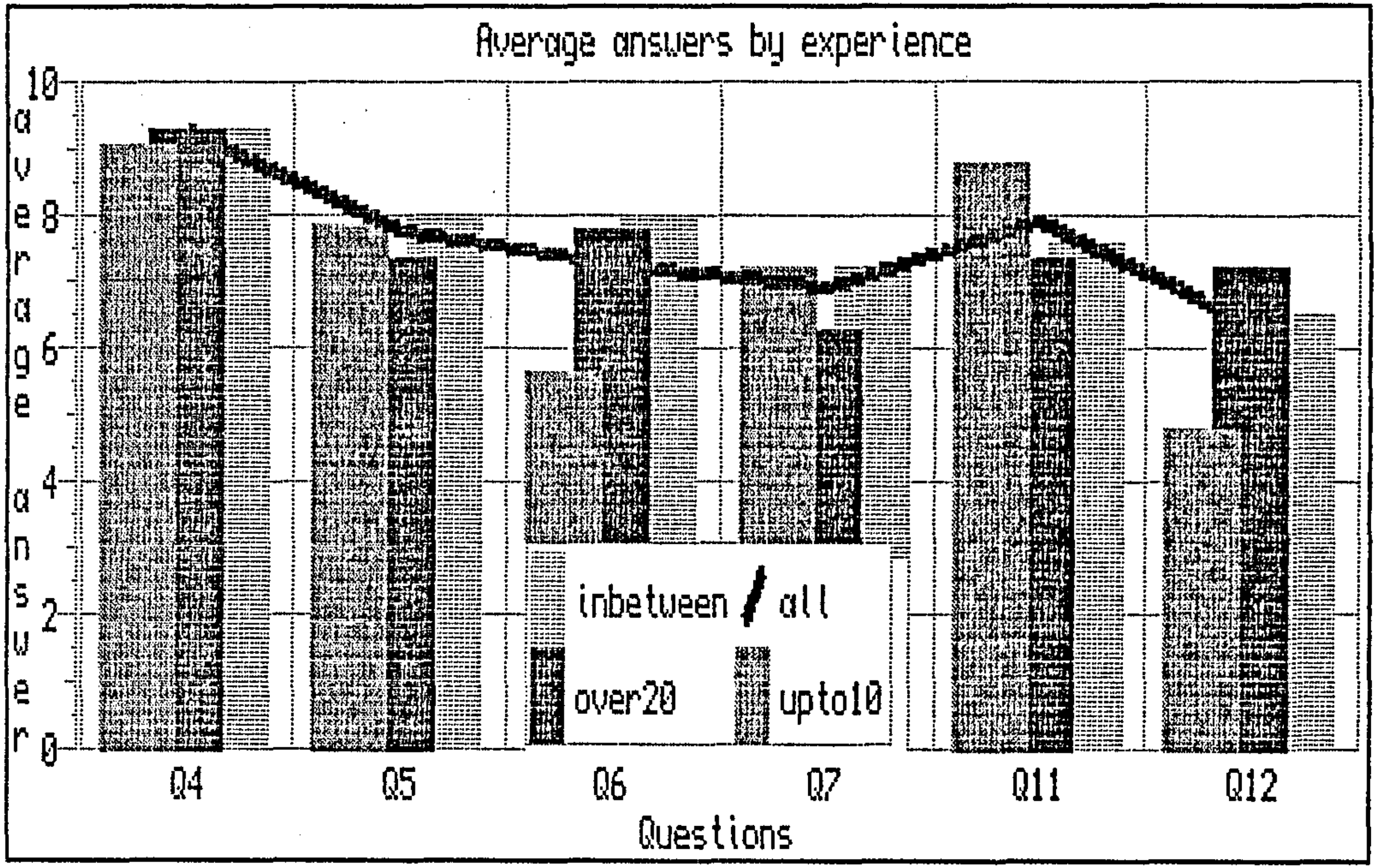
Graph 6.1


Average answers by occupation
average answer
10
8
6
4
2
0
Q4
Q5
Q6
Q7
Q11
Q12
Questions
design
prod
research
machine
tech
all

Graph 6.2


Average answer by age
average answer
0
2
4
6
8
10
Q4
Q5
Q6
Q7
Q11
Q12
Questions
in20s
in40s
all
in30s
over49

Graph 6.3



Average answers by experience
average answer
0
2
4
6
8
10
Q4
Q5
Q6
Q7
Q11
Q12
Questions
inbetween
all
over20
upto10

to be manufactured during design.

The view is strongly held that process plans have considerable, but not total, influence on the way components are manufactured. Some difference of opinion is apparent, as machinists consider process plans to have significantly more influence than either designers or technicians believe. Production engineers and researchers take a view falling between the two. On analysis by age, those aged fifty or over appear significantly less convinced that process plans influence the way components are manufactured.

Generally, respondents are convinced of their competence to compare the three process plans. The production engineers are conspicuously more confident than others, with designers and researchers much less so. Not surprisingly, younger or less experienced respondents have a lower estimation of their competence than the rest.

On average the three components are regarded as being similar, though perhaps more so in manufacturing than design terms. The most experienced and oldest respondents consider the components to be less similar.

When asked to express how familiar the wording used in plan C was to them, the average response indicates considerable familiarity. Designers and researchers indicate almost total familiarity, which is in marked contrast to the production engineers who find it much less familiar. The wording in plan C is also noticeably more familiar to the younger and least experienced respondents.

Finally, and perhaps somewhat obviously, a trend is indicated in the response to question twelve, asking about the importance of using a traditional style in process planning. With increasing age or experience, tradition is considered to be more important. By occupation, machinists consider a traditional style very important, whereas researchers consider it relatively unimportant. Interestingly, designers also take a below average view of the importance of using traditionally styled process plans.

6.3 Analysis of preferences

In the context of this discussion, 'preferences' refers to those questions which asked the respondent to rank the three process plans according to how helpful he found them based on certain criteria. Respondents were not allowed to rank the process plans equally and therefore a clear first, second and third preference was given in all answers. Again, age groups and experience groups have been rationalized to even out the sample sizes.

A scoring system was used to quantify the preferences, assigning points as follows:-

(a) Three points for first preference

(b) Two points for second preference

(c) One point for last preference

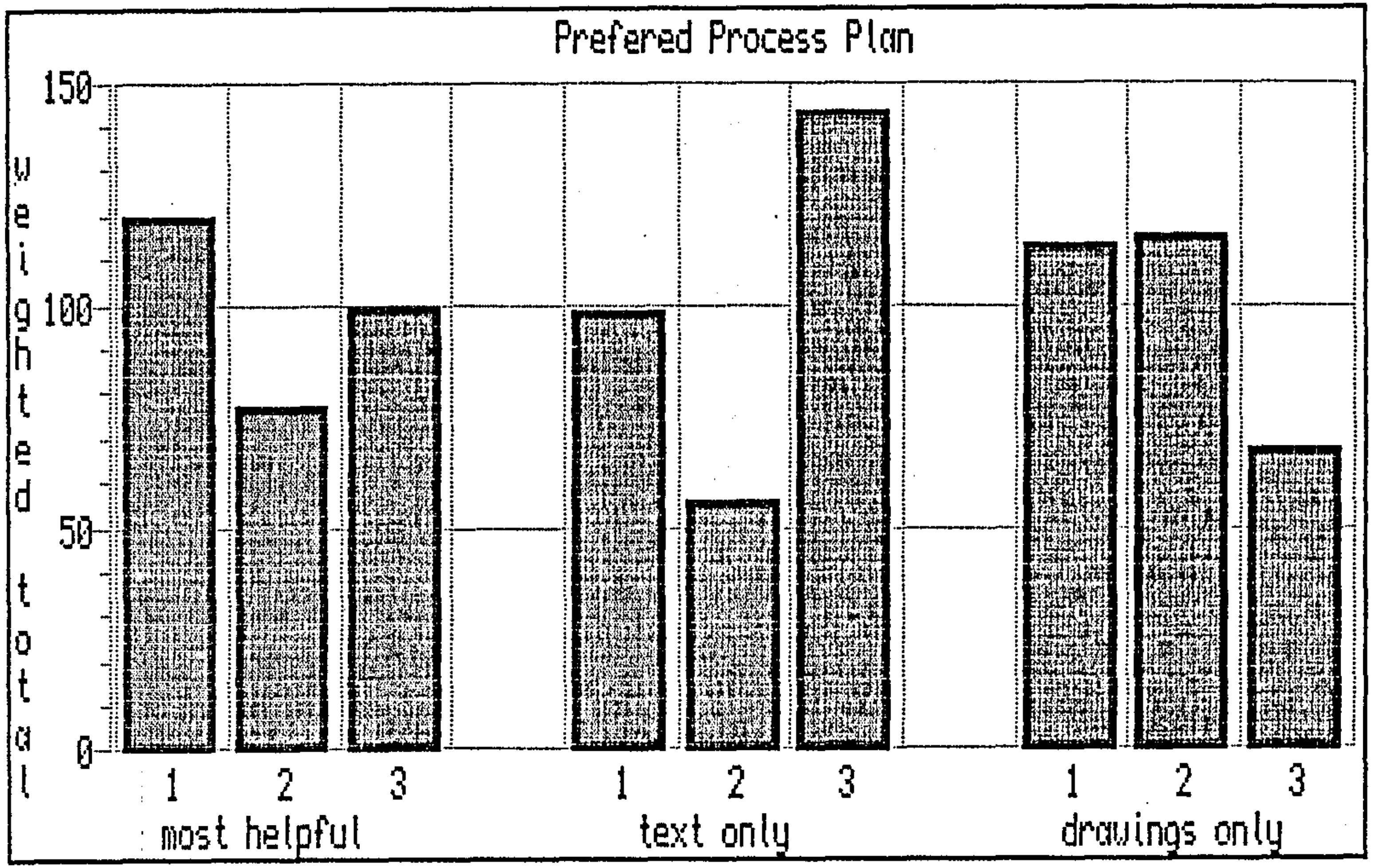
By scoring the answers of all respondents in this way graph 6.4 was produced, indicating the total weighted preferences. In order to analyse the results by occupation, age and experience, preferences were quantified in the same way for each of these groups, then divided by the size of the group to give weighted average preferences (see graphs 6.5-7).

Process plan A is generally considered to be the most helpful of the three, with plan C second and plan B last. On analysing the answers by occupation, technicians rank plan B second in preference to plan C, while machinists consider plan C to be the most helpful of all, with plan A second and plan B third.

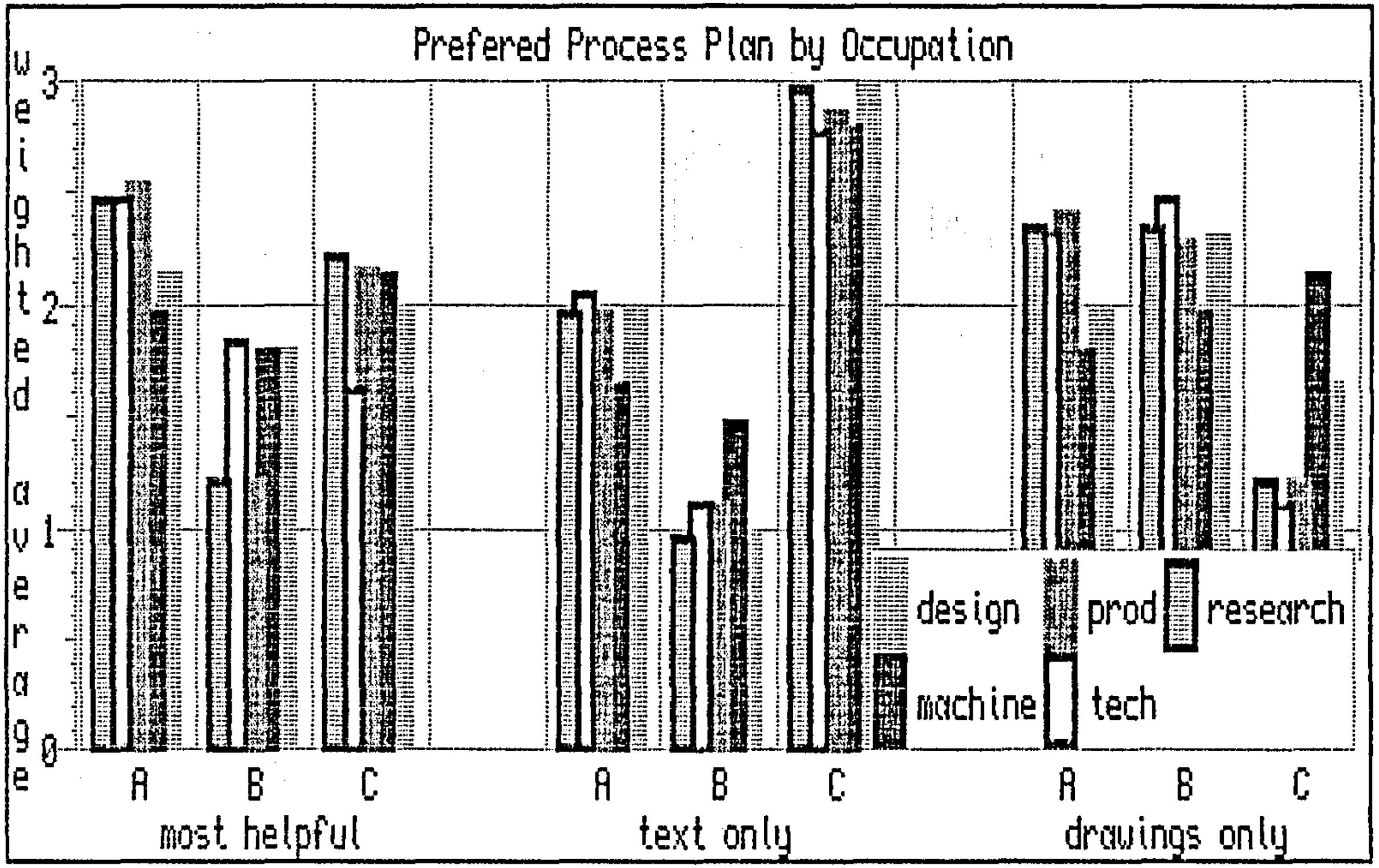
Discrimination between the process plans is less extreme by designers and machinists, with production engineers and researchers having the strongest views.

The preferences by age and experience are more consistent with the general result, but the fifty and over age group do

Graph 6.4

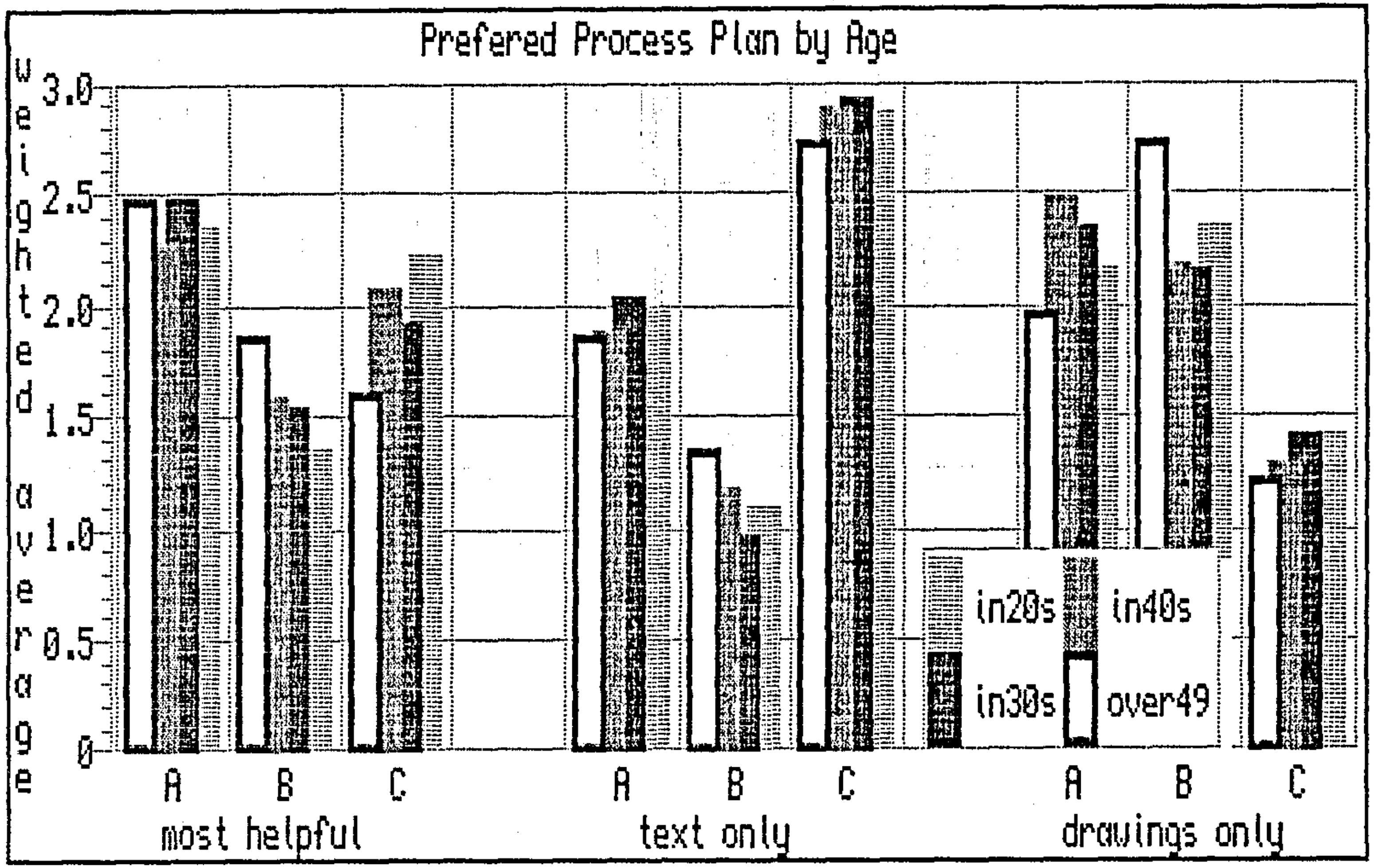

Prefered Process Plan
weighted total
150
100
50
0
1
2
3
most helpful
text only
drawings only

Graph 6.5

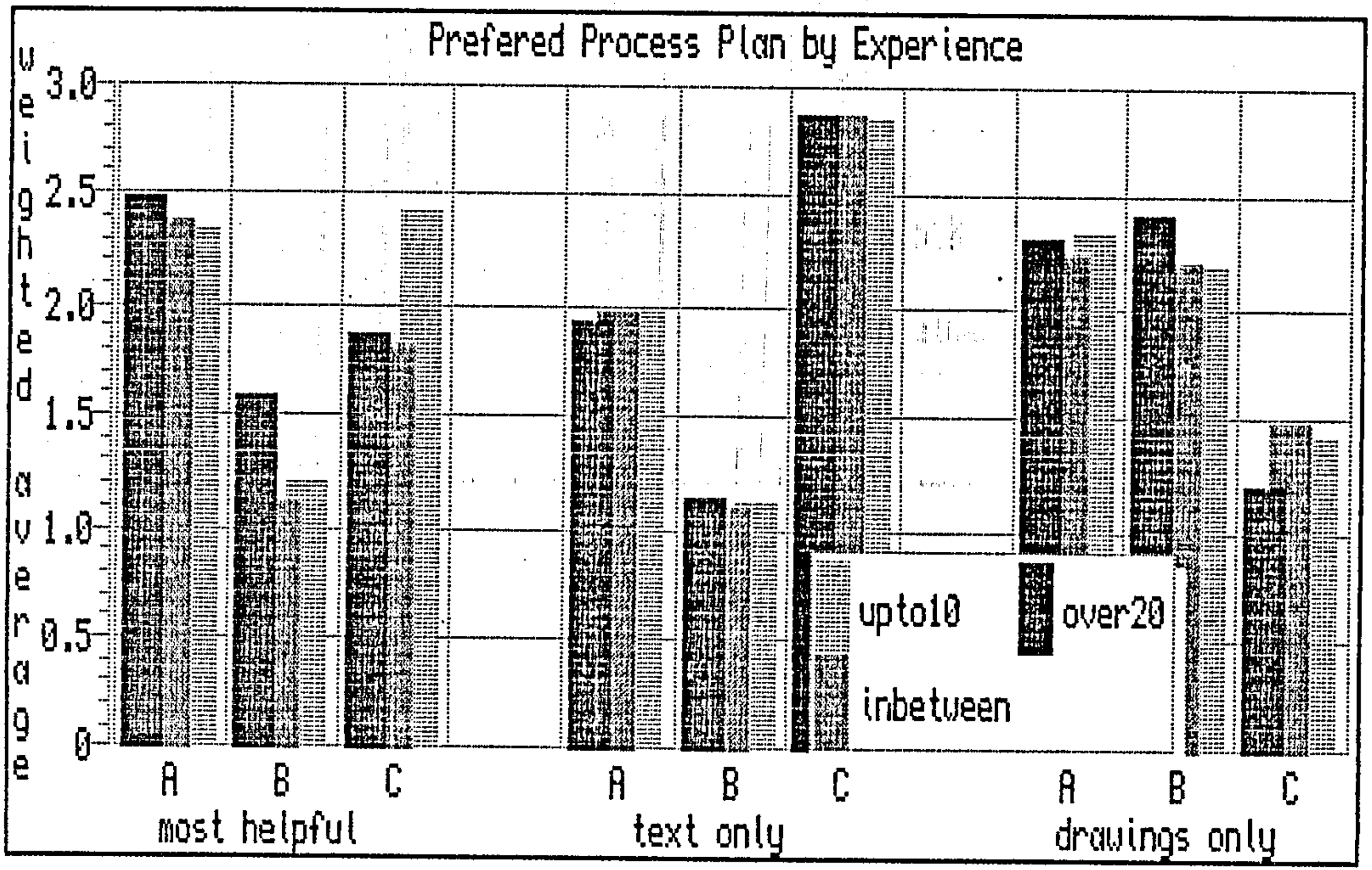


Prefered Process Plan by Occupation
weighted average
0
1
2
3
A
B
C
most helpful
text only
drawings only
design
prod
research
machine
tech

Graph 6.6


Prefered Process Plan by Age
weighted average
3.0
2.5
2.0
1.5
1.0
0.5
0
A
B
C
most helpful
A
B
C
text only
A
B
C
drawings only
in20s
in40s
in30s
over49

Graph 6.7



Prefered Process Plan by Experience
weighted average
3.0
2.5
2.0
1.5
1.0
0.5
0
A B C
most helpful
A B C
text only
A B C
drawings only
upto10
over20
inbetween

consider plan B to be more helpful than plan C. It is also notable that those with least experience marginally prefer plan C to plan A and rank plan B last of all. However, with increasing experience, plan C appears to lose popularity in favour of plan B.

When asked to consider the same question with the drawings removed, that is text only, the order of preference changes significantly. Process plan C is considered to be much more helpful than either plan A or B, with B least helpful of all. This overall pattern does not change when analysed by occupation, age or experience. The level of discrimination between plans A and B is less between machinists than other occupations, and discrimination generally decreases with increasing age.

On comparing the three process plans by drawings alone (text removed), plan C proves the least helpful, with little to choose between plans A or B for the most helpful. However, there are certain anomolies, with designers finding plan B distinctly more helpful than plan A, whereas machinists actually find plan C more helpful than either plan A or B. This is a surprising result because it is impossible to manufacture the component shown in plan C from the drawings alone! Perhaps there is a hidden message here.

Age has minimal effect on preference, though the fifty and over age group do show a strong preference for plan B that is well above the average. This is mirrored to a lesser extent by the most experienced respondents, who also show a slight preference for plan B.

6.4 Analysis of comments

Naturally it was not possible to quantify the comments made by subjects during and after completing the questionnaire, however all comments were recorded. Many responded with detailed and constructive comments, while others had little to say.

Comments were recorded independently of the answers given by subjects in the questionnaire. The only parameter used to indicate the source of the comment was the occupation of the individual concerned, to preserve anonymity.

The question about the importance of considering how components are to be manufactured during design prompted more comments than any other question. Several subjects referred to problems they had personally experienced because of lack of communication between the 'drawing office' and the shop-floor. In particular, two NC programers from different companies complained about the lack of awareness designers had for manufacturing problems. A common source of problems being tolerances which were too tight and the absence of features to hold the work with while machining. On further investigation, it usually transpired that the designer could relax the tolerances without detriment to the design, and add features to clamp onto without difficulty, only he just was not aware of the need. In one of these companies, the design office was actually situated on another site, approximately forty miles away.

Opinions about the relationship between designers and manufacturers were not all bad, in one instance a process planner stated that their department had a very good relationship with the drawing office and spoke of considerable cooperation between them on many jobs. In the same company though, a machinist complained of poor communication with the planning department, so clearly there is more than one important link in the design to manufacture chain. None of the designers ventured comments on their relationships with process planners.

On the importance of using a traditional style in process planning, one of the process planners observed that familiarity with the style of the process plan tended to reduce scrap, and that this was an argument for retaining a traditional style.

Many comments were made that have a direct bearing on the attitude towards plan C and the familiar engineering language therein. One machine operator suggested that there was too much detail in plan C for him, but that it would be beneficial to less experienced operators. Several others took a different view, which is summed up in the words of one technician who said that it was best to give as much detail as possible in a process plan, as it is easier to ignore excess information than to request further explanation. A few process planners feared that the level of detail in plan C would be insulting to an experienced machinist but this view was not supported by any of the machinists.

A draughtsman disparagingly compared plan C to model kit assembly instructions, adding that he felt 2D drawings were better. By way of contrast, a technician suggested that it was unwise to presume that everybody will understand engineering drawings, especially some of the specialized symbols used by the drawing office.

Prompted by the question regarding the wording of plan C, a production engineer commented that some older (imperial) drawings appear on the shop-floor from time to time, so even if usually metric it was wise to specify explicitly that the dimensions were metric using the 'mm' postfix. An NC programer was encouraged by the use of radial dimensioning which he thought would be helpful when programing certain types of NC machines that do not use diametral measurement.

One of the technicians commented in detail about the terminology used in plan C. He found the description of the facing operation unusual and preferred the term turn, also preferring the word 'spigot' to 'step'. He also connected the word 'recess' with an operation on a face, not a diameter, for which he expected to see the word 'undercut' instead. In fact, apart from the word spigot, all of them are equally acceptable synonyms within the familiar engineering language for simple turning.

Another technician observed that the format of plan C was similar, in some respects, to something he had encounterd before when working on a government rehabilitation scheme to introduce non-engineers to the terminology and concepts of manufacturing. He added that this was very useful, although later on trainees were taught to use conventional engineering drawings. Several others commented that if the text of plan C were combined with the drawing of plan A it would form an ideal process plan.

The remaining comments covered a range of points, including the observation that there were important details missing from all of the process plans. A researcher expressed disappointment at the absence of batch sizes and machine specifications, and a technician commented on the absence of work holding, fixturing or tooling information. All of which are valid comments, but in the introduction to the questionnaire it was stated that the examples were extracts, not complete process plans.

A machinist observed that plan B uses ISO hole/shaft tolerancing codes, and suggested that those unfamiliar with the system would not be able to interpret the drawing. A case for the specification of surface finish in the process plans was argued by an NC programer who gave the example of a turned aluminium component he once had to program. The component had to be plasma sprayed after turning, but at first it wouldn't 'take' because the surface finish was too smooth, and therefore a rougher than normal finish had to be achieved using an increased feed rate.

Finally, one of the designers who thought he had an insight into the purpose of the questionnaire suggested that defining a design in terms of the way it should be manufactured would not necessarily verify that the design is good.

6.5 Summary of findings

If the composition of the sample group was not ideal in a statistical sense, there was compensation in the enthusiasm and expertise of the respondents. Although a detailed statistical analysis is precluded, the findings constitute a valuable indicator of opinions that might be formalized in future work.

The indications are that there is considerable awareness and support for the need to design for manufacture, unfortunately the evidence of it's practise is not so strong. Moreover, the design to process planning relationship is not the only important link in the design to manufacture chain, the link must be established all the way through to the shop-floor.

Apart from the oldest and most experienced, there appears to be an encouragingly open-minded attitude to changing the style of process plans, if justified, although more effort would be needed to build up the confidence of machinists. Process plans are not yet regarded as definitive manufacturing instructions, but there is variation in their influence from company to company.

Although it is difficult to generalize, the popular misconception that too much detail in a process plan is insulting to the skilled machinist appears unfounded. What is more, there are clear indications that the younger and less experienced welcome more detail, up to the point of the step by step drawings used in plan C, which were not otherwise well received. A more likely reason for lack of detail might be evident from a casual comment expressed by one process planner, that plan C was very nice but he wouldn't like to have to write it.

Improvements in the familiar engineering language are suggested from the comments made by respondents. For instance, the inclusion of the 'mm' metric postfix, an increased range of manufacturing synonyms and the capacity to specify surface finish requirements.

To the extent that it can be tested within the questionnaire, there appears to be substantial merit in the use of familiar engineering language as a means of communicating manufacturing information. The sample of familiar engineering language used in plan C was generally very well understood, with the highest average response from designers. Which would appear to suggest that familiar engineering language would be equally useful as an input language to a design based system. The least experienced respondents also found the language easy to understand and therefore, as a means of interpreting manufacturing information by a wide range of individuals, it would be ideal.

7 FUTURE WORK AND CONCLUSIONS

Clearly more work is needed on the turning example to expand the range of operations and to rectify the shortcomings that have been highlighted in it. A high priority must also be to fully integrate the language with the solid modeller and produce a tru ly interactive system. Otherwise, the system can only remain of academic interest. To accomplish this it would be necessary to obtain the subroutine driven version of BOXER, complete with the capacity to include tolerancing information, as soon as it is available.

Given the problems associated with the system hardware on which the application program, GILT and BOXER are presently implemented, an alternative system would be needed to improve the accuracy and speed of execution. This would enable typical engineering components to be modelled without impractically long delays.

The outcome of the questionnaire inspires further work in this area, but it would be important that a close relationship with practising engineers is maintained to guarantee the validity of the resulting familiar engineering language. To this end, a comprehensive and extensive interviewing programme might be beneficial in quantifying the opinions and attitudes of engineers with statistical certainty. From the outcome of this research many of the outstanding questions regarding the standardization of terminology, scope and content of familiar engineering language could be resolved with complete confidence.

Although the particular example chosen to illustrate the technique and method of application of familiar engineering language is concerned with turning, the approach is sufficiently general to be appropriate for a wide range of engineering processes. The technique of using the command processor, method of integration with the solid modeller via the application program and mechanics of graph design would all be the same. Only the content of the language graphs would differ substantially from that of the turning example already described.

Underlying the definition of any familiar engineering language map, will be the same basic principles governing the complexity of the resulting language map, as well as the repetition of sub-constructions (patterns) within and between language graphs. The repeating patterns that emerged in the familiar engineering language graphs for turning will also appear in the graphs for other processes. For example, the subgraph for tolerancing is independent of the type of engineering process and would be as appropriate in the familiar engineering language map of milling, grinding or drilling as it is in turning. Similar patterns will also occur in the sub-constructions for the dimensioning of operations, although there would be differences in the terminology.

The CSG solid modeller used in the example, has already been shown to be appropriate for the modelling of components produced by metal cutting processes. A significant proportion of manufacturing effort could be addressed immediately, by extending the familiar engineering language approach to other metal cutting processes. Among the processes that would be of interest are the following:-

(a) Drilling
(b) Milling
(c) Grinding

Subsequent work on milling operations has already been undertaken (78,79). Although familiar language graphs have not yet been defined, indications are that milling may be harder to define than turning because of the potentially complex orientations of the tool/workpiece (compound angles). Moreover, a convention for referencing features on milled components is not readily apparent. In the case of turning, a systematic referencing procedure is obvious.

Drilling by contrast is a comparatively simple process, both geometrically and in engineering terms, implying that the corresponding familiar language graphs would be of a lower order of complexity. In drilling, fewer dimensions are necessary to specify operations which are also quite limited in variety.

Grinding can be further subdivided into two main processes, the first being surface grinding and the second rotational grinding. While surface grinding has much in common geometrically with milling, rotational grinding has much in common with turning.

The geometric similarities between rotational grinding and turning operations are such that in many cases the same familiar engineering language and solid modeller commands can be used. In the example, although this ability has not previously been mentioned, it is possible to use the entire language map to describe rotational grinding operations by using the general term 'machine' instead of the explicit term 'turn'. By simply adding the atom 'grind' to all states containing the atom 'turn' in the example familiar language map, explicit reference would also be possible.

Other reasons may now be clear for the absence of process specific operations like 'parting off' in the example familiar language map, in order to preserve generality. Consider the following examples:-

FACE TO A LENGTH OF 1ØØ.Ø

FACE GRIND TO A LENGTH OF 1ØØ.Ø

TURN THE OUTSIDE DIAMETER OF THE BAR DOWN BY 1Ø.Ø

GRIND THE OUTSIDE DIAMETER OF THE BAR DOWN BY Ø.1

The principle difference between turning and rotational grinding operations is one of tolerance, whereas a tight tolerance on a diameter suggests that a finish grinding operation is necessary, otherwise turning is usually sufficient. The consideration of the characteristic surface finishes produced by the two processes is frequently subordinate to the importance of dimensional accuracy. For example, the following operations are identical apart from their tolerances, consequently the choice of process, either turning or grinding, will be determined entirely by the accuracy that can be attained:-

MACHINE THE BAR DOWN TO A DIAMETER OF 4Ø.ØØ WITHIN A TOLERANCE OF 39.98 TO 4Ø.Ø3

MACHINE THE BAR DOWN TO A DIAMETER OF 4Ø.ØØ WITHIN A TOLERANCE OF 39.99 TO 4Ø.ØØ

This raises a fundamentally important issue with respect to the use of familiar engineering language to describe multiple processes, in the detail design of components, that must be resolved. Should machining operations be specified by dimensional requirements alone, or is it important to specify processes too. If full consideration is to be given to the principle of design for manufacture, then designers have to be aware of the manufacturing processes implied by their choice of tolerances. If the general term 'machine' is used then the implications of creating an additional manufacturing operation may be missed, whereas if process specific operations are used, like 'turn' and 'grind', then to achieve tighter tolerances the designer is forced to specify two seperate operations and cannot miss the significance of tolerancing. Therefore, the process specific approach would appear preferable.

To make the process specific approach work, the respective characteristics of different processes would have to be encoded in order to ensure that sensible processes are selected, for example:-

TURN THE BAR DOWN TO 25 PLUS OR MINUS Ø.ØØ5

GRIND THE BAR DOWN BY 5Ø PLUS OR MINUS Ø.5

In the general case where the process is not explicitly stated (machine....), neither of the above operations could be considered wrong, although clearly incorrect as presented.

The successful implementation of these suggestions for future work would facilitate the development of a familiar engineering language driven design system capable of wide industrial application. The outcome of such work can not yet be judged, however, many important conclusions can be drawn from the research that has already been completed. These conclusions are fundamental to the familiar engineering language approach irrespective of the engineering processes involved.

Computer integrated manufacture is widely believed to be important for the future success of the manufacturing industry in the western world, and implies changes in manufacturing organizations from the machine-shop to the boardroom. One aspect of computer integration that is fundamental to CIM is establishing a link between CAD and CAM. In this respect, the relationship between design and process planning is vital, highlighting, among other things, the need to design for manufacture.

The influence of design on manufacture can be seen throughout the three major stages: specification, conceptual and detail design. Implicit within the design specification are basic manufacturing constraints dictated by quantity and quality requirements. During conceptual design, and subsequently in detail design, the choice of material, tolerances and surface finishes all have a direct influence on the method, and therefore cost of manufacture. It is not surprising, to find that multidisciplinary design teams, representing all facets of design and production, generate profitable designs.

The arguments in favour of designing for manufacture are compelling, they would also appear to be widely appreciated and acknowledged throughout the engineering industry. Unfortunately, evidence suggests that this awareness has not been translated into practice with anything like the same conviction.

The reasons for this are many fold, and include past and present prejudices within organizations, inappropriate organizational structures and physical separation of departments. For instance, it is hard to see how design for manufacture can be practised when the drawing office is situated forty miles from the factory site, which proved to be the case in one of the companies at which questionnaires were administered. At another company, managers proudly boasted of their achievement in improving design for manufacture, by moving the drawing office one floor closer to the production engineering department.

The prejudices between departments are frequently substantial barriers to design for manufacture, aided and abetted by inappropriate organizational structures that foster separation, and not integration, of design and production personnel. It would also appear that in concentrating on the most obvious link in the design for manufacture chain, between design and process planning, that the equally vital link between process planning and manufacture, is overlooked. Certainly there is evidence from the results of the questionnaire that the process plan is not regarded with anything like the degree of importance required to implement design for manufacture successfully.

These considerations, it may be argued, are not relevant in relation to CIM, where ultimately many of these 'personnel' problems will be eliminated, as the personnel themselves are eliminated through computerization and automation. However, it is naive to assume that such a transition can happen overnight, and experience has shown that it is hard to achieve complete electronic integration without first understanding the problems of full manual integration. Nevertheless, as a powerful tool for improving information and communication, the computer can be instrumental in bringing about the necessary changes.

One of the aspects in which computers can be helpful is 3D solid modelling, which has the potential to integrate all aspects of manufacturing through a single comprehensive product description, in terms that are as relevant to the designer as they are to the NC programer. Considerable reductions in the lead time from design to manufacture are also possible through the increased efficiency of reusing, rather than recreating, information about a product as it passes through the system. An example of this is the 2D engineering drawing, which as the primary means of communicating design information for subsequent use by production engineers, has created an artificial interface between design and manufacture. The reality of manufacturing a component in 3D is all to often different from the ability to draw it in 2D, resulting in delays while problems are solved.

CSG based solid modellers are particularly appropriate for modelling engineering machining processes such as turning, because they enable close simulation of machining operations. By using the boolean difference operation, metal removal from a predefined solid, representing the stock material, can be emulated. However, solid modelling is still at an early stage and several important engineering considerations must be accommodated, like tolerancing, before it can be a credible substitution for the 2D engineering drawing.

Tolerancing is of fundamental importance in engineering, for without information to describe the accuracy and to an extent the surface characteristics of a component, it cannot be manufactured. Very recent work at Leeds University suggests that a tolerancing facility is shortly to be available within BOXER, which should overcome these problems.

The alliance of computers with the concepts of standardization, classification and coding makes it feasible to rationalize the chaotic mass of information inherent in manufacturing organizations. Perhaps the best example of the power of these management tools, is the success of Group Technology in rationalizing engineering batch manufacture. It follows that the application of standardization, classification and coding as a methodology for structuring manufacturing information may be a prerequisite in unifying CAD and CAM, which in the final analysis is really a series of data processing operations. The creation and management of large design and manufacturing databases is simplified if the information is well structured. Undoubtedly, such large databases will be necessary in integrating design with process planning.

Process planning has come in for close scrutiny recently, and is seen by many as the keystone necessary to bridge the gap between CAD and CAM. The crucial nature of process planning has led to a growth in numbers and use of computer aided process planning systems.

Four classes of process planning system have been identified: variant, variant-generative, semi-generative and generative. Variant systems can be compared with word processors, improving the speed and efficiency with which process plans can be produced, but have no influence on their content. Variant-generative systems are typically the result of attempts to enhance basically variant systems by adding some generative features. The change in emphasis towards a creative approach is first evident within semi-generative systems, and distinguishes them from the variant style. However, the level of manual data input and supervision is such that they cannot really be described as generative. Generative systems, including expert systems, attempt to synthesize a process plan from first principles every time, and are believed by many to be the only means of integrating CAD and CAM.

Despite a certain partial success with generative systems, for a limited range of processes or components, to date no one has been successful in producing a fully generative process planning system. A number of researchers take the view that there are substantial obstacles to the success of the generative approach. Difficulties lie in the automatic recognition of machined surfaces from a 3D solid model, the selection of primary processes and optimum manufacturing sequences.

Although recent work on the development of expert process planning systems has renewed optimism, the success of expert systems to date has been in limited problem domains, not in broad problem domains like process planning. The risk exists that the necessary expertise will continue to evade researchers for some time. Possibly more significantly, our understanding of the manufacturing environment in general is still very limited and thus the capacity to encode it, in whatever form, must be similarly limited.

The culmination of these arguments leads to the proposition that a fundamental change in the present approach to design and process planning may be beneficial to wider manufacturing objectives. This new approach takes into account the aspirations of design for manufacture and generative process planning, but avoids many of the problems so far associated with them.

It has been proposed that the designer who is already designing for manufacture in all senses, must in fact moot an outline process plan for the components he designs. The 2D engineering drawing is an inadequate and incomplete means of communicating the designers efforts in this respect. Therefore, instead of passing on the outline process plan it is in fact wasted only to be duplicated at a later stage during process planning.

For the designer who does not yet practise design for manufacture, there is little in the traditional approach to encourage him to change. To bring such change about, the design approach itself must be changed to make design for manufacture explicit rather than implicit practice.

In both of these cases a vehicle for encouraging and capturing design for manufacture information, in addition to geometric information, is required. A technique has been proposed, using a 3D solid modelling system that captures design and manufacturing information together and encourages the user to be aware of manufacturing considerations.

The success of this approach depends on the ability to describe manufacturing information that is predominately textual, in conjunction with geometric information. Therefore, a communication interface is called for which, to ensure the widest possible user familiarity for input or output purposes, best consists of familiar engineering language.

Having considered several formats for familiar engineering language: a rigid structure, keyword searching and natural language programing, a hybrid format was chosen. A hybrid format overcomes some of the problems associated with the other formats, like non-specific words in keyword searching or the difficulty of interpreting ungrammatical and agrammatical sentences in natural language. The resulting familiar engineering language is rigidly structured but flexible because it is extensive.

The command processor, GILT, aids the development and implementation of familiar engineering language by simplifying language construction, filtering out unreasonable input and facilitating customization to accommodate new 'dialects'. The potential also exists to generate highly descriptive output suitable for process planning purposes.

Using the graphical approach proposed, assisted by the graph drawing, GILT input and graph expansion utilities, the development of familiar engineering language graphs is simplified. The implementation of familiar engineering language, though potentially problematical, has been demonstrated to be possible through the example of familiar engineering language applied to simple turning.

Based on the experience of defining a relatively small sample of familiar engineering language for simple turning operations it is not possible to draw definitive conclusions, however, certain fundamental relationships are evident:-

(a) A familiar engineering command contains four basic elements, whether implied of left to standard defaults, these elements are always present:-

OPERATION + LOCATION + DIMENSIONS + TOLERANCES

(b) Patterns, or sub-constructions, are constantly repeated throughout familiar engineering language graphs, which if recognized simplify language construction.

(c) The permutations of a familiar engineering language command are proportional to the number of dimensions specified by it, increasing in steps of approximately ten to a power four. In the case of turning, third order graphs like 'reces' will generate around ten to a power twelve permutations.

(d) The size of a familiar engineering graph is not directly related to the number of permutations generated by it.

These findings are evidence of the feasibility of the familiar engineering language approach, because enormous freedom of expression can be achieved without resulting in prohibitively large or ambiguous language structures. Thus the development of a familiar engineering language for all turning operations and other production processes is possible, given the necessary resources.

A degree of standardization is beneficial in the design of familiar engineering language, if consistent and correct practice is to be encouraged. In some respects, the example of familiar engineering language for simple turning operations is not sufficiently standardized and allows excessive variety of expression. For instance, with hindsight, the capacity to specify radial dimensions is of doubtful value and not common engineering practice. In other respects, it is too standardized, for example in assuming that surface finish information is redundant. This assumption is not corroborated by the results of the questionnaire.

In contrast, several of the assumptions made were not contested, for instance the assumption that the terms roughing and finishing are redundant. The use of logical defaults in dimensioning has also proved useful, enabling users to choose between detailed or succinct commands. For output purposes, the assumption that detailed commands are preferable to succinct commands is also supported by the findings of the questionnaire. Detailed machining instructions were widely appreciated by the machinists questioned, somewhat contradicting the view held by several process planners that too much detail in a process plan would insult the skilled machinist. This misconception is perhaps too easily accepted by the process planner who, understandably, does not wish to find himself extra work.

Before applying familiar engineering language on a wider scale, further work would be beneficial in reviewing the standards to be observed within it. One standard that is very clear, is the the need to specify the nature of the operation before any dimensions, tolerances or location, otherwise commands become awkwardly phrased.

Another area in which standardization is important is the referencing of features on components, for example when an operation is to be performed on, or relative to, a previously machined feature. Then the considerable potential for ambiguity dictates that a convention for describing features is used. In the example of familiar engineering language for simple turning, diameters are automatically renumbered from right to left after each operation, and can be referenced in this manner. The problem of referencing features is likely to be exacerbated in the case of prismatic components, which do not have the intrinsically sequential relationship that turned features share.

In conclusion it can be stated that the integration of design and manufacture can be approached in a new way using familiar engineering language. The results of research into this approach so far are encouraging and indicate the potential to further develop the approach for industrial application.

APPENDICES

APPENDIX A Experimental questionnaire 1

This questionnaire was designed to simulate the use of familiar engineering language, by asking respondants to annotate a series of drawings detailing the manufacture of a simple turned component.

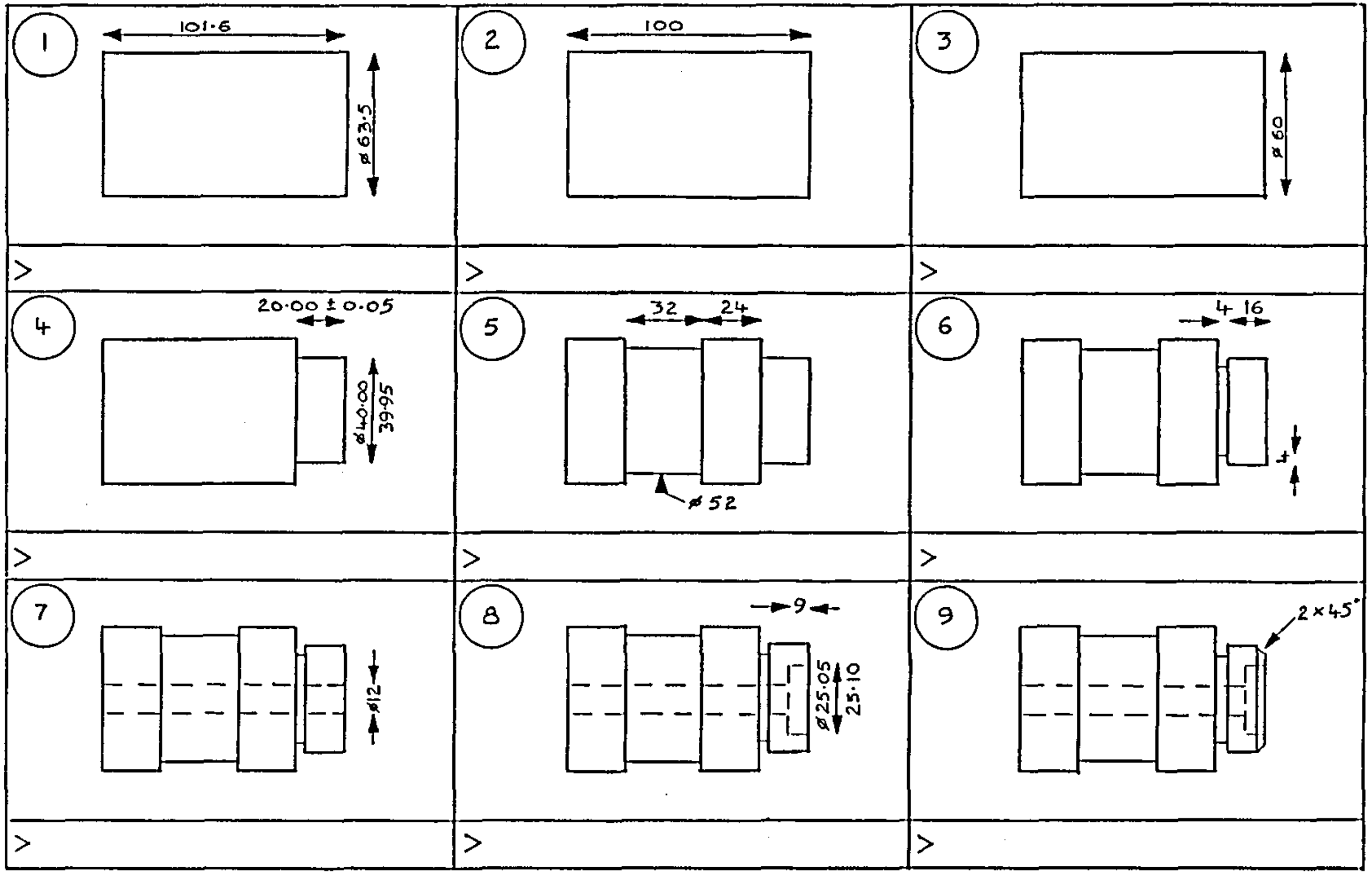
EXERCISE 1

Annotate a diagrammatical process plan with familiar engineering language

On the next page is a diagrammatical process plan for a simple turned component. The diagrams are numbered from 1 to 9 and represent the individual operations in the sequence required to produce it.

If each diagram shows the condition to be achieved by a single operation, add suitable text to accompany it.

Assume that work holding, tooling, speeds and feeds are determined automatically, therefore, you need not specify them in the text.


1
101·6
ø63·5
2
100
3
ø60
4
20·00 ± 0·05
ø40·00
39·95
5
32
24
ø52
6
4
16
4
7
ø12
8
9
ø25·05
25·10
9
2 × 45°

APPENDIX B Experimental questionnaire 2

This questionnaire was designed to test the output from a familiar engineering language driven design system, in the form of a process plan. The process plan was then compared with two other 'typical' process plans.

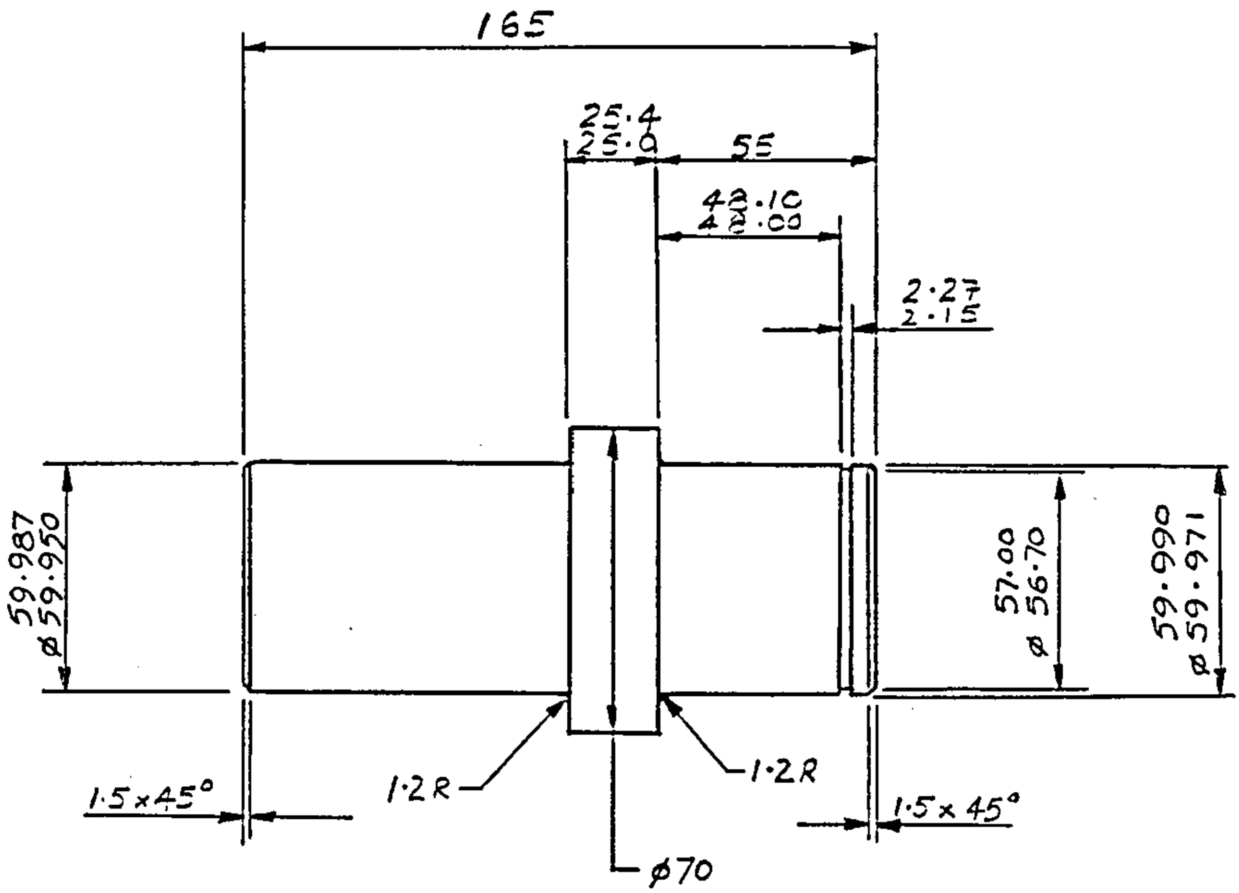
EXERCISE 2

<u>Compare three different styles of process planning</u>

On the following pages are extracts from three different process plans for simple turned components. Although none of the components are exactly alike, they are sufficiently similar to enable a comparison of the syle and content of their respective process plans.

After a careful study of each of the process plans, refer to the questions section where you are asked to give answers which reflect the impressions you have formed of them.

Please answer all the questions, if the exact reply you wish to give is not possible then choose one which is closest to it and use the 'any comments' section to explain further.



PROCESS MASTER

Description

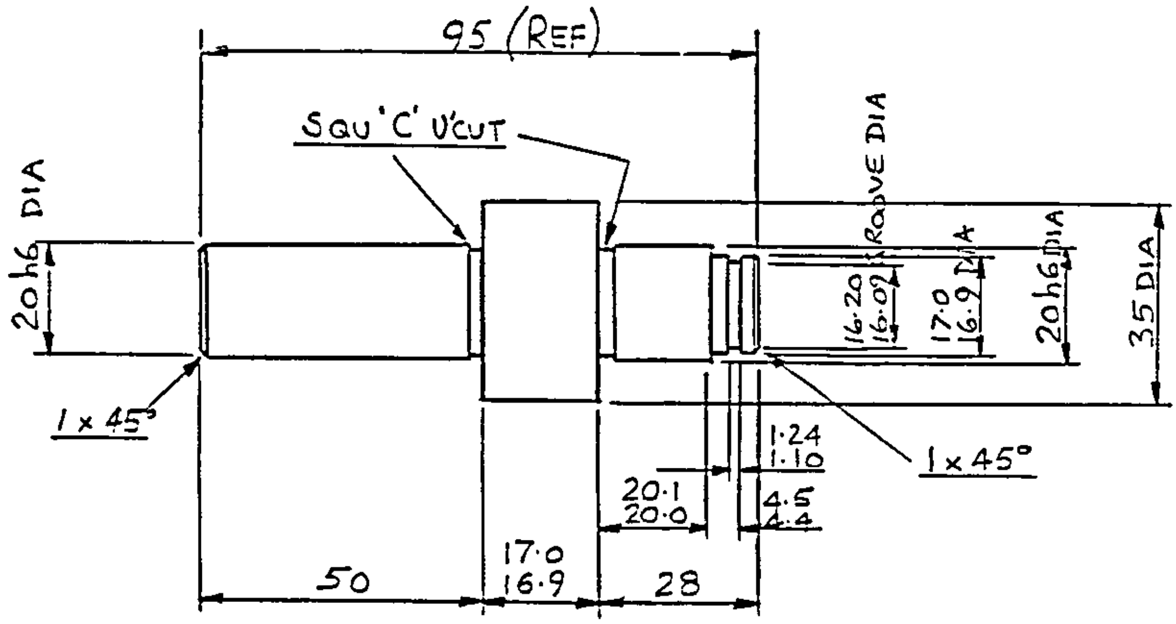
12.5T Runner Pin

Operation 1

Face Centre Turn 59.990/59.971 Leave Grinding Allowance .25 Chamfer Groove 2.27/2.15 Wide X 57.00/56.70 Diameter Part Off to 165 lg.

Operation 2

Hold in collets Turn 59.987/59.950 Leave Grinding Allowance .25 face Chamfer & Centre



Job Specification

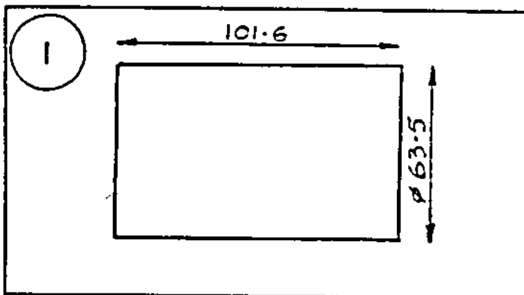
Description

BRT MS ROD BS970 220 MO7 35 DIA TOL SMALL H10

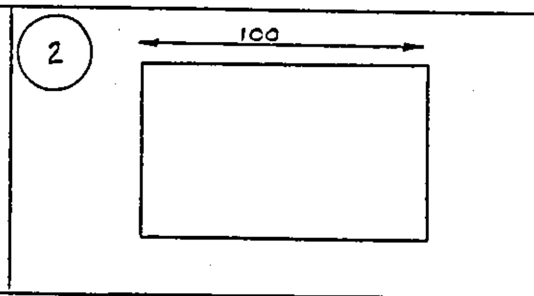
Operation Description

Turn face U/cut Groove Chamfer and Part off CPTE

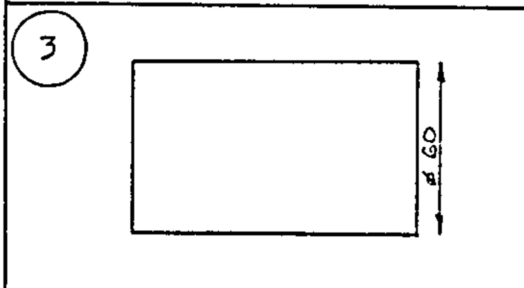
Remove sharp corners



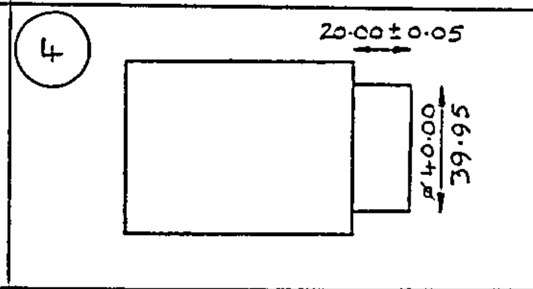
Stock material round bar 63.5 diameter by 101.6 long



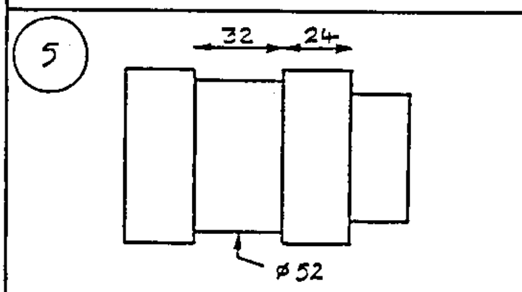
Face the length down to 100



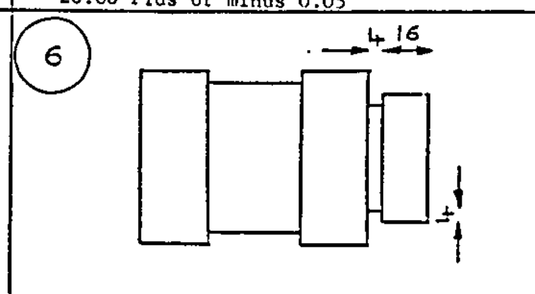
Turn the outside diameter of the bar down to 60



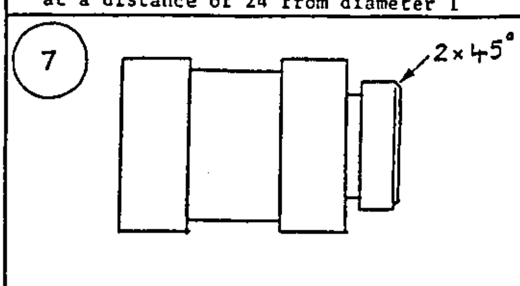
Turn a step in the bar having diameter 40 tolerance 40.00 to 39.95 and length 20.00 Plus or minus 0.05



Turn a recess in diameter 2 with dimensions of width 32 and diameter 52 at a distance of 24 from diameter 1



Turn an undercut at a distance of 16 from the end having width 4 and depth 4



Chamfer the end of the bar width 2 at an angle of 45°

QUESTION EXAMPLES

On the following pages there are ten questions for you to answer, but in some instances the manner of reply demanded may not be familiar to you, so please examine the two examples given below:-

(a) How well can you swim? Mark on the following scale:-

```
                 Ø          5          1Ø
like a brick  | . . . . | X . . . |  like a fish
```

(b) Rank the three major political parties according to how likely you think they are to win the next election:-

1..Conservative........	1..Alliance.............
2..Labour..............	2.......................
3..Alliance............	3..Conservative/Labour..

Now please complete the questionnaire overleaf.

(1) What is your age?......

(2) Occupation? (eg designer/process planner/turner..)

....

(3) How competent are you to machine the components shown in the process plans? Mark your ability on the following scale:-

	Ø		5		10	
novice	|	. . . .	|	. . . .	|	expert

(4) In machining terms how do you think the components compare? Mark the degree of similarity on the following scale:-

	Ø		5		10	
opposite	|	. . . .	|	. . . .	|	identical

(5) 'Process plans should follow a traditional style'. Indicate how much you agree with this statement:-

	Ø		5		10	
totally	|	. . . .	|	. . . .	|	not at all

(6) Rank the process plans according to how helpful you would find them if you had to machine the components yourself:-

1...............

2...............

3...............

(7) Rank the process plans according to how helpful you would find them if the text was removed (ie drawings only):-

1...............

2...............

3...............

(8) Rank the process plans according to how helpful you would find them if the drawings were removed (ie text only):-

1...............

2...............

3...............

(9) Do you think the process plans could be improved in any way and if so how?

....

....

(10) Any other comments?

....

....

APPENDIX C Results of questionnaire

The results of the fifty correctly completed questionnaires are listed. They are in no particular order, simply the order in which they were administered.

	Q1	Q2	Q3	Q4	Q5	Q6
1)	3	5	4	8.5	8	9
2)	2	5	3	9.5	7.5	5.5
3)	3	5	5	9	8	3
4)	3	5	4	8.25	9.25	8.5
5)	2	5	4	1Ø	5	2.5
6)	2	5	3	9	9	8
7)	2	5	2	9	9	2
8)	3	5	4	9	9	5
9)	5	4	6	1Ø	5	1Ø
1Ø)	5	4	6	1Ø	6	8
11)	2	4	4	1Ø	5	Ø
12)	2	4	4	1Ø	8	5
13)	6	4	6	7	5	9.75
14)	3	4	5	1Ø	7	9
15)	3	4	5	1Ø	7	7.5
16)	5	4	6	1Ø	5	8
17)	4	4	6	1Ø	6	1Ø
18)	6	1	6	1Ø	8	7
19)	2	3	4	9	7	4
2Ø)	4	3	6	1Ø	7	9
21)	3	4	5	1Ø	7	9
22)	5	1	6	7.5	7	3
23)	2	3	4	1Ø	9	9
24)	4	3	6	1Ø	9	9
25)	2	2	4	9	8	6

	Q1	Q2	Q3	Q4	Q5	Q6
26)	4	3	6	8	1Ø	1Ø
27)	2	3	4	8	7.5	7
28)	3	3	5	1Ø	1Ø	7.5
29)	2	3	5	1Ø	7.5	1Ø
3Ø)	2	4	5	1Ø	8	8
31)	3	2	5	1Ø	1Ø	8
32)	4	2	5	1Ø	1Ø	8
33)	2	3	5	1Ø	6	9
34)	4	3	6	1Ø	9	8
35)	3	3	6	1Ø	9.5	8.75
36)	2	1	4	8	9	8.5
37)	4	1	6	9	9	2
38)	2	2	5	1Ø	6	5
39)	3	3	5	1Ø	8	9
4Ø)	4	4	6	1Ø	8	5
41)	5	4	6	9	6	9
42)	5	3	6	1Ø	1Ø	1Ø
43)	3	3	5	8	8	1Ø
44)	4	4	6	7	9	7
45)	3	2	5	1Ø	1Ø	9
46)	3	2	5	2	1Ø	9
47)	3	3	5	9	9	9.75
48)	2	3	5	1Ø	7	9
49)	3	1	5	1Ø	5	5
5Ø)	4	1	6	1Ø	5	7

	Q7	Q8	Q9	Q1Ø	Q11	Q12
1)	8.25	132	123	312	1Ø	4
2)	6.75	213	213	312	8.5	5
3)	9.5	132	123	312	1Ø	2.5
4)	9.25	312	312	312	9.5	1.5
5)	7	132	123	312	1Ø	3
6)	8.5	312	213	312	1Ø	5
7)	9.25	132	213	312	7	5
8)	5	312	213	312	9	5
9)	6	312	213	312	1Ø	5
1Ø)	8	213	213	213	8	1Ø
11)	2	123	213	312	1Ø	5
12)	1Ø	312	123	312	1Ø	1Ø
13)	3.75	213	123	312	1Ø	5
14)	7	123	123	312	Ø	7
15)	2.5	213	213	312	6	5
16)	9	312	321	312	1Ø	5
17)	7	132	213	312	7	5
18)	5	123	213	312	1Ø	8
19)	7	312	123	312	1Ø	3
2Ø)	7.5	312	213	312	9	9
21)	7	132	213	312	1Ø	5
22)	6	132	213	312	9	8
23)	8	312	213	123	1Ø	5
24)	8	312	123	312	1Ø	8
25)	8	312	312	321	8	7

	Q7	Q8	Q9	Q10	Q11	Q12
26)	7	132	123	312	2	9
27)	9	132	213	312	2	7
28)	10	312	123	312	10	10
29)	8.5	123	123	312	3	2
30)	1.5	123	123	312	10	7
31)	9	312	213	312	10	8
32)	7	231	231	231	9	10
33)	8	132	123	312	10	9
34)	8	132	123	312	8	9
35)	6	132	213	312	4	6.5
36)	4	312	321	312	9.75	2.5
37)	8	213	123	312	10	8
38)	8	321	231	312	10	10
39)	8	132	312	312	10	5
40)	6	213	123	312	3	7
41)	8	123	213	312	10	8
42)	1	123	213	321	1	10
43)	7	213	213	312	0	7
44)	6	132	123	312	4	9
45)	10	123	132	312	10	10
46)	10	123	312	312	9	10
47)	6.5	132	123	312	1.5	5
48)	8.25	312	321	312	10	0
49)	3	231	213	312	9	5
50)	4	312	312	312	9	1

APPENDIX D Graph input language for turning example

The turning example is referred to in chapter 5. In this listing of the graph input language tabulations and underscoring has been added to give extra clarity.

```
ATOMS
-----

    VAL 3

ENDATOMS

DEF REFER 1
===========

    STATE 1
    --------
       STOCK              -            2

    STATE 2
    --------
       NL                 -            3

    STATE 3
    --------
       'TURN'             1            5
       'MACHINE'          1            5
       'RECESS'           1            14
       'FACE'             1            4
       'UNDERCUT'         1            14
       'FINISH/ED'        1 2          15
       'STOP'             1 2          15
       'END'              1 2          15
       'QUIT'             1 2          15

    STATE 4
    --------
       FACE               -            2

    STATE 5
    --------
       'A'                1            7
       'AN'               1            8
       'RECESS'           1            9
       'UNDERCUT'         1            9
       -                  -            6

    STATE 6
    --------
       STEP               -            2
       LINEA              -            2

    STATE 7
    --------
       'RECESS'           1            9
       STEP               -            2

    STATE 8
    --------
       'UNDERCUT'         1            9
```

```
    STATE 9
    --------
       'IN'            1           10
       -               3           14

    STATE 10
    --------
       'DIAMETER'      1           11
       'THE'           1 3         12
       'BAR'           1 3         14
       'OUTSIDE'       1 3         13

    STATE 11
    --------
       VAL             4           14
       -               3           14

    STATE 12
    --------
       'BAR'           1           14
       'OUTSIDE'       1           13
       -               -           13

    STATE 13
    --------
       'DIAMETER'      1           14

    STATE 14
    --------
       RECES           -           2

    STATE 15
    --------
       NL              -           0

END
```

```
DEF STOCK 2
===========

     STATE 1
     --------
        'STOCK'          1             2
        'MATERIAL'       1             3

     STATE 2
     --------
        'MATERIAL'       1             4
        -                -             4

     STATE 3
     --------
        'STOCK'          1             5
        -                -             4

     STATE 4
     --------
        'A'              1             5
        -                -             5

     STATE 5
     --------
        'ROUND'          1             8
        'CIRCULAR'       1             6

     STATE 6
     --------
        'CROSS'          1             7
        'SECTION'        1             8
        -                -             8

     STATE 7
     --------
        'SECTION'        1             8

     STATE 8
     --------
        'BAR'            1             9

     STATE 9
     --------
        'WITH'           1             10
        'HAVING'         1             10
        'OF'             1             12
        -                -             13

     STATE 10
     --------
        'DIMENSIONS'     1             11
        'SIZE'           1             13
        -                -             13
```

STATE 11

'OF'	1	13
-	-	13

STATE 12

'SIZE'	1	13
'DIMENSIONS'	1	13
-	-	13

STATE 13

'DIAMETER'	1 3	14
'RADIUS'	1 4	14
'LENGTH'	1 5	15
'WIDTH'	1 6	15
VAL	2	30

STATE 14

VAL	2	16

STATE 15

VAL	2	23

STATE 16

TOLER	7	17
-	8	17

STATE 17

'BY'	1	18
'X'	1	18
'AND'	1	18
-	-	18

STATE 18

'LENGTH'	1 5	19
'WIDTH'	1 6	19
VAL	2	20

STATE 19

VAL	2	21

STATE 20

'LONG'	1 5	21
'WIDE'	1 6	21
TOLER	-	22
-	5 8 9	0

```
STATE 21
--------
   TOLER          7 9         0
   -              8 9         0

STATE 22
--------
   'LONG'         1 5 7 9     0
   'WIDE'         1 6 7 9     0
   -              5 7 9       0

STATE 23
--------
   TOLER          7          24
   -              8          24

STATE 24
--------
   'BY'           1          25
   'X'            1          25
   'AND'          1          25
   -              -          25

STATE 25
--------
   'DIAMETER'     1 3        26
   'RADIUS'       1 4        26
   VAL            2          27

STATE 26
--------
   VAL            2          28

STATE 27
--------
   'DIAMETER'     1 3        28
   'RADIUS'       1 4        28
   TOLER          -          29
   -              3 8 9       0

STATE 28
--------
   TOLER          7 9         0
   -              8 9         0

STATE 29
--------
   'DIAMETER'     1 3 7 9     0
   'RADIUS'       1 4 7 9     0
   -              3 7 9       0

STATE 30
--------
   TOLER          -          31
   -              -          32
```

```
STATE 31
--------
   'DIAMETER'     1 3 7       17
   'RADIUS'       1 4 7       17
   'LONG'         1 5 7       24
   'WIDE'         1 6 7       24
   -              3 7         17

STATE 32
--------
   'DIAMETER'     1 3         16
   'RADIUS'       1 4         16
   'LONG'         1 5         23
   'WIDE'         1 6         23
   -              3 8         17

END
```

```
DEF LINEA 3
===========

    STATE 1
    --------
       'TO'           1 2         11
       'DIAMETER'     1 2 5       2
       'RADIUS'       1 2 6       2
       'BAR'          1 2         5
       'THE'          1 2         3
       'OUTSIDE'      1 2         4
       'DOWN'         1 2         6
       VAL            1 3         20

    STATE 2
    --------
       VAL            4           5
       -              -           5

    STATE 3
    --------
       'BAR'          2           5
       'DIAMETER'     2 5         5
       'RADIUS'       2 6         5
       'OUTSIDE'      2           4

    STATE 4
    --------
       'DIAMETER'     2 5         5

    STATE 5
    --------
       'TO'           2           11
       'DOWN'         2           10
       'BY'           2 7         14
       'X'            2 7         14

    STATE 6
    --------
       'DIAMETER'     2 5         7
       'RADIUS'       2 6         7
       'BAR'          2           10
       'THE'          2           8
       'OUTSIDE'      2           9
       -              -           10

    STATE 7
    --------
       VAL            4           10

    STATE 8
    --------
       'BAR'          2           10
```

```
    'DIAMETER'      2 5             10
    'RADIUS'        2 6             10
    'OUTSIDE'       2               9

STATE 9
--------
    'DIAMETER'      2 5             10

STATE 10
--------
    'TO'            2               11
    'BY'            2 7             14
    'X'             2 7             14
    -               7               14

STATE 11
--------
    VAL             3               12
    'RADIUS'        2 6             17
    'DIAMETER'      2 5             17
    'DEPTH'         2 7             17
    'A'             2               16
    -               -               16

STATE 12
--------
    'RADIUS'        2 6 8           19
    'DIAMETER'      2 5 8           19
     TOLER          -               13
     -              8 10 13          0

STATE 13
--------
    'DIAMETER'      2 5 8 9 13 0
    'RADIUS'        2 6 8 9 13 0
    -               8 9 13           0

STATE 14
--------
    VAL             3 8             19
    'A'             2               15
    -               -               15

STATE 15
--------
    'DEPTH'         2               17

STATE 16
--------
    'RADIUS'        2 6             18
    'DIAMETER'      2 5             18
    'DEPTH'         2 7             17

STATE 17
--------
    'OF'            2               18
```

```
   -                -               18

STATE 18
--------
   VAL              3 8             19

STATE 19
--------
   TOLER            9 13             0
   -                10 13            0

STATE 20
--------
   TOLER            11              21
   -                -               21

STATE 21
--------
   'OFF'            2               22

STATE 22
--------
   'DIAMETER'       2 5             23
   'RADIUS'         2 6             23
   'THE'            2               24
   -                -               24

STATE 23
--------
   VAL              4 8 12 13 0
   -                -               27

STATE 24
--------
   'BAR'            2               25
   'DIAMETER'       2 5             27
   'RADIUS'         2 6             27
   'OUTSIDE'        2               26

STATE 25
--------
   'DIAMETER'     2 5 8 12 13        0
   'RADIUS'       2 6 8 12 13        0

STATE 26
--------
   'DIAMETER'       2 5             27

STATE 27
--------
   'OF'             2               28
   -                8 12 13         0

STATE 28
--------
   'THE'            2               29
```

```
              -                  -          29

           STATE 29
           --------
              'BAR'              2 8 12 13   0

       END
```

```
DEF FACE 4
==========

    STATE 1
    --------
       'TO'            1 2          17
       'DIAMETER'      1 2          2
       'RADIUS'        1 2          2
       'BAR'           1 2          8
       'THE'           1 2          4
       'LENGTH'        1 2          5
       'END'           1 2          5
       'X'             1 2 5        22
       'BY'            1 2 5        22
       'DOWN'          1 2          9
       'OFF'           1 2          11
       VAL             3 5          24

    STATE 2
    --------
       VAL             4            3

    STATE 3
    --------
       'TO'            2            17
       'DOWN'          2            16
       'BY'            2 5          22
       'X'             2 5          22

    STATE 4
    --------
       'BAR'           2            8
       'LENGTH'        2            5
       'END'           2            5

    STATE 5
    --------
       'OF'            2            6
       -               -            8

    STATE 6
    --------
       'THE'           2            7
       -               -            7

    STATE 7
    --------
       'BAR'           2            8

    STATE 8
    --------
       'TO'            2            17
       'OFF'           2            16
```

```
   'DOWN'          2           16
   'BY'            2 5         22
   'X'             2 5         22

STATE 9
--------
   'DIAMETER'      2           10
   'RADIUS'        2           10
   -               -           11

STATE 10
--------
   VAL             4           16

STATE 11
--------
   'BAR'           2           16
   'THE'           2           12
   'LENGTH'        2           13
   'END'           2           13
   'TO'            2           17
   'BY'            2 5         22
   'X'             2 5         22

STATE 12
--------
   'BAR'           2           16
   'LENGTH'        2           13
   'END'           2           13

STATE 13
--------
   'OF'            2           14
   -               -           16

STATE 14
--------
   'THE'           2           15
   -               -           15

STATE 15
--------
   'BAR'           2           16

STATE 16
--------
   'TO'            2           17
   'BY'            2 5         22
   'X'             2 5         22
   -               2           22

STATE 17
--------
   VAL             3 6         18
   'A'             2           20
   'LENGTH'        2           21
```

```
STATE 18
--------
   'LONG'       2         23
   TOLER        7         19
   -            8 11      0

STATE 19
--------
   'LONG'       2 11       0
   -            11         0

STATE 20
--------
   'LENGTH'     2         21

STATE 21
--------
   'OF'         2         22
   -            -         22

STATE 22
--------
   VAL          3 6       23

STATE 23
--------
   TOLER        7 11       0
   -            8 11       0

STATE 24
--------
   TOLER        9         25
   -            -         25

STATE 25
--------
   'OFF'        2         26

STATE 26
--------
   'THE'        2         27
   -            -         27

STATE 27
--------
   'LENGTH'     2         28
   'END'        2         28

STATE 28
--------
   'OF'         2         29
   -            -         31

STATE 29
--------
```

```
         'THE'           2              30
         'DIAMETER'      2              32
         'RADIUS'        2              32
         -               -              30

      STATE 30
      --------
         'BAR'           2 6 1 0 11 0

      STATE 31
      --------
         'DIAMETER'      2              32
         'RADIUS'        2              32

      STATE 32
      --------
         VAL             4 6 1 0 11 0

END
```

```
DEF STEP 5
==========

    STATE 1
    --------
       'STEP'          1 2         2
       'SHOULDER'      1 2         2

    STATE 2
    --------
       'IN'            2           3
       -               -           7

    STATE 3
    --------
       'DIAMETER'      2           4
       'RADIUS'        2           4
       'THE'           2           5
       -               -           5

    STATE 4
    --------
       VAL             4           7

    STATE 5
    --------
       'BAR'           2           7
       'OUTSIDE'       2           6
       -               -           6

    STATE 6
    --------
       'DIAMETER'      2           7

    STATE 7
    --------
       'TO'            2           8
       'WITH'          2           8
       'HAVING'        2           8
       'OF'            2           10
       -               -           11

    STATE 8
    --------
       'DIMENSIONS'    2           9
       'SIZE'          2           11
       'A'             2           11
       -               -           11

    STATE 9
    --------
       'OF'            2           11
       -               -           11
```

```
STATE 10
--------
   'SIZE'          2           11
   'DIMENSIONS'    2           11

STATE 11
--------
   'DIAMETER'      2 5         12
   'RADIUS'        2 6         12
   'DEPTH'         2 7         12
   'LENGTH'        2 8         21
   'WIDTH'         2 8         21
   VAL             3           30

STATE 12
--------
   VAL             3 9         13

STATE 13
--------
   TOLER           10          14
   -               11          14

STATE 14
--------
   'BY'            2           15
   'X'             2           15
   'AND'           2           15
   -               -           15

STATE 15
--------
   VAL             3 8 9       18
   'LENGTH'        2 8         16
   'WIDTH'         2 8         16

STATE 16
--------
   VAL             3 9         17

STATE 17
--------
   TOLER           10 12       0
   -               11 12       0

STATE 18
--------
   TOLER           10          19
   -               11          20

STATE 19
--------
   'LONG'          2 12        0
   'WIDE'          2 12        0
   -               12          0
```

STATE 20

'LONG'	2	17
'WIDE'	2	17
-	12	0

STATE 21

VAL	3 9	22

STATE 22

TOLER	10	23
-	11	23

STATE 23

'BY'	2	24
'X'	2	24
'AND'	2	24
-	-	24

STATE 24

VAL	3	27
'DIAMETER'	2 5	25
'RADIUS'	2 6	25
'DEPTH'	2 7	25

STATE 25

VAL	3 9	26

STATE 26

TOLER	10 12	0
-	11 12	0

STATE 27

TOLER	-	28
-	-	29

STATE 28

'DIAMETER'	2 5 9 10 12	0
'RADIUS'	2 6 9 10 12	0
'DEEP'	2 7 9 10 12	0
-	8 9 10 12	0

STATE 29

'DIAMETER'	2 5	26
'RADIUS'	2 6	26
'DEEP'	2 7	26

```
    -              8 9 11 12   0

STATE 30
--------
    TOLER          -           31
    -              -           32

STATE 31
--------
    'DEEP'         2 7 9 10    14
    'DIAMETER'     2 5 9 10    14
    'RADIUS'       2 6 9 10    14
    'LONG'         2 8 9 10    23
    'WIDE'         2 8 9 10    23
    -              5 9 10      14

STATE 32
--------
    'DEEP'         2 7 9       13
    'DIAMETER'     2 5 9       13
    'RADIUS'       2 6 9       13
    'LONG'         2 8 9       22
    'WIDE'         2 8 9       22
    -              5 9 11      14

END
```

DEF RECES 6
===========

STATE 1

'AT'	1 2	7
'BETWEEN'	1 2	12
-	1 17	2

STATE 2

'TO'	2	3
'WITH'	2	3
'HAVING'	2	3
'OF'	2	5
-	-	6

STATE 3

'DIMENSIONS'	2	4
'SIZE'	2	6
-	-	6

STATE 4

'OF'	2	6
-	-	6

STATE 5

'SIZE'	2	6
'DIMENSIONS'	2	6

STATE 6

RECSA	-	0

STATE 7

'A'	2	8
-	-	10

STATE 8

'DISTANCE'	2	9

STATE 9

'OF'	2	11
-	-	11

STATE 10

```
        'DISTANCE'       2            9
        -                -           11

     STATE 11
     --------
        RECSB            -            0

     STATE 12
     --------
        RECSC            -            0

END
```

```
DEF RECSA 7
===========

    STATE  1
    --------
       'DIAMETER'     2 5         2
       'RADIUS'       2 6         2
       'DEPTH'        2 7         2
       'LENGTH'       2 8        28
       'WIDTH'        2 8        28
       VAL            3          37

    STATE  2
    --------
       VAL            3 9         3

    STATE  3
    --------
       TOLER          10          4
       -              11          4

    STATE  4
    --------
       'BETWEEN'      2          18
       'FROM'         2          18
       'BY'           2           5
       'X'            2           5
       'AND'          2           5
       -              -           5

    STATE  5
    --------
       VAL            3 8 9       8
       'LENGTH'       2 8         6
       'WIDTH'        2 8         6

    STATE  6
    --------
       VAL            3 9         7

    STATE  7
    --------
       TOLER          10         11
       -              11         11

    STATE  8
    --------
       TOLER          10          9
       -              -          10

    STATE  9
    --------
       'LONG'         2          11
```

```
    'WIDE'          2           11
    -               -           11

STATE 10
--------
    'LONG'          2            7
    'WIDE'          2            7
    -               11          11

STATE 11
--------
    'AT'            2           12
    'BY'            2           12
    'X'             2           12
    'AND'           2           12
    -               -           12

STATE 12
--------
    'A'             2           13
    -               -           15

STATE 13
--------
    'DISTANCE'      2           14

STATE 14
--------
    'OF'            2           16
    -               -           16

STATE 15
--------
    'DISTANCE'      2           14
    -               -           16

STATE 16
--------
    VAL             3 12        17

STATE 17
--------
    TOLER           10          23
    -               11          23

STATE 18
--------
    VAL             3 13        19

STATE 19
--------
    TOLER           10          20
    -               11          20

STATE 20
--------
```

```
   'TO'              2            21
   'AND'             2            21
   -                 -            21

STATE 21
--------
   VAL               3 14         22

STATE 22
--------
   TOLER             10           23
   -                 11           23

STATE 23
--------
   'FROM'            2            24
   -                 16 18         0

STATE 24
--------
   'SHOULDER'        2            27
   'DIAMETER'        2            27
   'THE'             2 16         25
   -                 16           25

STATE 25
--------
   'FREE'            2            26
   -                 -            26

STATE 26
--------
   'END'             2 18          0

STATE 27
--------
   VAL               4 15 18       0

STATE 28
--------
   VAL               3 9          29

STATE 29
--------
   TOLER             10           30
   -                 11           30

STATE 30
--------
   'BY'              2            31
   'X'               2            31
   'AND'             2            31
   -                 -            31

STATE 31
--------
```

```
   VAL              3            34
   'DIAMETER'       2 5          32
   'RADIUS'         2 6          32
   'DEPTH'          2 7          32

STATE 32
--------
   VAL              3 9          33

STATE 33
--------
   TOLER            10           11
   -                11           11

STATE 34
--------
   TOLER            -            35
   -                -            36

STATE 35
--------
   'DIAMETER'       2 5 9 10     11
   'RADIUS'         2 6 9 10     11
   'DEEP'           2 7 9 10     11
   -                9 10         11

STATE 36
--------
   'DIAMETER'       2 5          33
   'RADIUS'         2 6          33
   'DEEP'           2 7          33
   -                9 11         11

STATE 37
--------
   TOLER            -            38
   -                -            39

STATE 38
--------
   'DEEP'           2 7 9 10      4
   'DIAMETER'       2 5 9 10      4
   'RADIUS'         2 6 9 10      4
   'LONG'           2 8 9 10     30
   'WIDE'           2 8 9 10     30
   -                5 9 10        4

STATE 39
--------
   'DEEP'           2 7 9         3
   'DIAMETER'       2 5 9         3
   'RADIUS'         2 6 9         3
   'LONG'           2 8 9        29
   'WIDE'           2 8 9        29
   -                5 9 11        4
```

END

```
DEF RECSB 8
===========

     STATE  1
     --------
        VAL              3 12          2

     STATE  2
     --------
        TOLER            10            3
        -                11            3

     STATE  3
     --------
        'FROM'           2             4
        -                16            8

     STATE  4
     --------
        'SHOULDER'       2             7
        'DIAMETER'       2             7
        'THE'            2 16          5
        -                16            5

     STATE  5
     --------
        'FREE'           2             6
        -                -             6

     STATE  6
     --------
        'END'            2             8

     STATE  7
     --------
        VAL              4 15          8

     STATE  8
     --------
        'TO'             2             9
        'WITH'           2             9
        'HAVING'         2             9
        'OF'             2            11
        -                -            12

     STATE  9
     --------
        'DIMENSIONS'     2            10
        'SIZE'           2            12
        -                -            12

     STATE 10
     --------
```

```
    'OF'            2          12
    -               -          12

STATE 11
--------
    'SIZE'          2          12
    'DIMENSIONS'    2          12

STATE 12
--------
    'WIDTH'         2 8        13
    'LENGTH'        2 8        13
    'DEPTH'         2 7        23
    'DIAMETER'      2 5        23
    'RADIUS'        2 6        23
    VAL             3          33

STATE 13
--------
    VAL             3 9        14

STATE 14
--------
    TOLER           10         15
    -               11         15

STATE 15
--------
    'BY'            2          16
    'X'             2          16
    'AND'           2          16
    -               -          16

STATE 16
--------
    VAL             3          20
    'A'             2          17
    -               -          17

STATE 17
--------
    'RADIUS'        2 6        18
    'DIAMETER'      2 5        18
    'DEPTH'         2 7        18

STATE 18
--------
    VAL             3 5        19

STATE 19
--------
    TOLER           10 18       0
    -               11 18       0

STATE 20
--------
```

```
     TOLER              -            21
     -                  -            22

STATE 21
--------
     'RADIUS'           2 6 10 18     0
     'DIAMETER'         2 5 10 18     0
     'DEEP'             2 7 10 18     0
     -                  7 10 18       0

STATE 22
--------
     'RADIUS'           2 6 9        19
     'DIAMETER'         2 5 9        19
     'DEEP'             2 7 9        19
     -                  5 9 11 18     0

STATE 23
--------
     VAL                3            24

STATE 24
--------
     TOLER              10           25
     -                  11           25

STATE 25
--------
     'BY'               2            26
     'X'                2            26
     'AND'              2            26
     -                  -            26

STATE 26
--------
     VAL                3            30
     'A'                2            27
     -                  -            27

STATE 27
--------
     'LENGTH'           2 8          28
     'WIDTH'            2 8          28

STATE 28
--------
     VAL                3            29

STATE 29
--------
     TOLER              10 18         0
     -                  11 18         0

STATE 30
--------
     TOLER              -            31
```

```
    -                -           32

STATE 31
--------
    'LONG'           2 10 18      0
    'WIDE'           2 10 18      0
    -                10 18        0

STATE 32
--------
    'LONG'           2           29
    'WIDE'           2           29
    -                11 18        0

STATE 33
--------
    TOLER            -           34
    -                -           35

STATE 34
--------
    'WIDE'           2 8 9 10    15
    'LONG'           2 8 9 10    15
    'DIAMETER'       2 5 9 10    26
    'RADIUS'         2 6 9 10    26
    'DEEP'           2 7 9 10    26
    -                5 9 10      26

STATE 35
--------
    'WIDE'           2 8 9       14
    'LONG'           2 8 9       14
    'DIAMETER'       2 5 9       25
    'RADIUS'         2 6 9       25
    'DEEP'           2 7 9       25
    -                5 9 11      26

END
```

```
DEF RECSC 9
===========

     STATE  1
     --------
        VAL              3 13        2

     STATE  2
     --------
        TOLER            10          3
        -                11          3

     STATE  3
     --------
        'TO'             2           4
        'AND'            2           4
        -                -           4

     STATE  4
     --------
        VAL              3 14        5

     STATE  5
     --------
        TOLER            10          6
        -                11          6

     STATE  6
     --------
        'FROM'           2           7
        -                16         11

     STATE  7
     --------
        'SHOULDER'       2          10
        'DIAMETER'       2          10
        'THE'            2 16        8
        -                16          8

     STATE  8
     --------
        'FREE'           2           9
        -                -           9

     STATE  9
     --------
        'END'            2          11

     STATE 10
     --------
        VAL              4 15       11

     STATE 11
```

```
--------
   'TO'             2               12
   'AT'             2               12
   'BY'             2               19
   'X'              2               19
   'AND'            2               19
   -                -               19

STATE 12
--------
   VAL              3               13
   'A'              2               16
   -                -               16

STATE 13
--------
   TOLER            -               14
   -                -               15

STATE 14
--------
   'RADIUS'     2 6 9 10 18          0
   'DIAMETER'   2 5 9 10 18          0
   'DEEP'       2 7 9 10 18          0
   -            5  9 10 18           0

STATE 15
--------
   'RADIUS'       2 6 9             21
   'DIAMETER'     2 5 9             21
   'DEEP'         2 7 9             21
   -              5 9 11 18          0

STATE 16
--------
   'RADIUS'       2 6               17
   'DIAMETER'     2 5               17
   'DEPTH'        2 7               17

STATE 17
--------
   'OF'             2               18
   -                -               18

STATE 18
--------
   VAL            3 9               21

STATE 19
--------
   VAL              3               13
   'A'              2               20
   -                -               20

STATE 20
--------
```

```
     'DIAMETER'     2 5          18
     'RADIUS'       2 6          18
     'DEPTH'        2 7          18

  STATE 21
  --------
     TOLER          10 18         0
     -              11 18         0

END
```

```
DEF TOLER 10
===========

    STATE 1
    --------
       'TO'              1 2          2
       'WITH/IN'         1 2          2
       'GENERAL'         1 2 4        9
       'TOLERANCE'       1 2          6
       'PLUS'            1 2 5        10

    STATE 2
    --------
       'A'               2            3
       'TOLERANCE'       2            5
       'GENERAL'         2 4          7

    STATE 3
    --------
       'GENERAL'         2 4          4
       -                 -            4

    STATE 4
    --------
       'TOLERANCE'       2            5

    STATE 5
    --------
       'OF'              2            6
       -                 -            6

    STATE 6
    --------
       'PLUS'            2 5          10
       VAL               3 6          13

    STATE 7
    --------
       'TOLERANCE'       2            8

    STATE 8
    --------
       'PLUS'            2 5          10
       VAL               3 6          13
       -                 -            0

    STATE 9
    --------
       'TOLERANCE'       2            6

    STATE 10
    --------
       'OR'              2            11
```

```
    STATE 11
    --------
       'MINUS'          2          12
    STATE 12
    --------
       VAL              3 7        0
    STATE 13
    --------
       'TO'             2          12
END

FINISH
```

REFERENCES

(1) FENNELL E. How to set about rebuilding Britain's manufacturing base. The Daily Telegraph, Tuesday, May 6, 1986, p 12.

(2) KNILL B, MILLS R B, TERESKO J, WEIMER G. Integrated manufacture - America's competitive strategy. Production Engineering, Special Report, February, 1986, p 1-32.

(3) HEGLAND D E. CAD/CAM integration - key to the automatic factory. Production Engineering, August, 1981, p 31-35.

(4) HEGLAND D E. Putting the automated factory together. Production Engineering, June, 1982, p 57-64.

(5) HALEVI G. The role of computers in manufacturing processes. John Wiley & Sons, 1980.

(6) COOK P, CORBETT J, PUGH S, WEIGHTMAN D. A guide to design for production. The Institute of Production Engineers, 1984.

(7) BRITISH STANDARDS INSTITUTION. Manual of British standards in engineering drawing and design. Hutchinson, 1984, chap 2.

(8) TURNER B T. Creative approaches to engineering design. The Chartered Mechanical Engineer, vol 22, no 10, November, 1975, p 85-89.

(9) PUGH S, SMITH D G. The dangers of design methodology. Proceedings of 'Changing Design', Portsmouth, 1976.

(10) HYDE W F. Improving productivity by classification, coding and database standardiazation - the key to maximizing CAD/CAM and group technology. Marcel Dekker, 1981.

(11) BURBIDGE J L. The introduction of group technology. Heinemann, 1975, p 219-231.

(12) LARDNER J F. Start today for the factory of tomorrow. Production Engineering, June, 1982, p 50-52.

(13) SCHAFFER G. Implementing CIM. American Machinist, Special Report 736, August, 1981, p 152-174.

(14) SCHAFFER G. GT via automated process planning. American Machinist, May, 1980, p 119-122.

(15) REQUICHA A A G. Representations for rigid solids: theory, methods, and systems. Computing Surveys, vol 12, no 4, December 1980, p 437-464.

(16) CAREY G C, PENNINGTON A. A study of the interface between CAD and CAM using geometric modelling techniques. Department of Mechanical Engineering, University of Leeds.

(17) JARED G E M. Introduction to solid modelling. Conference proceedings of the British Computer Society, London, 26th February, 1986.

(18) SABIN M. 15 years of geometric modelling. Conference proceedings of the British Computer Society, London, 26th February, 1986.

(19) WOODWARK J R. Solid Modelling - the set-theoretic approach. Conference proceedings of the British Computer Society, London, 26th February, 1986.

(20) MEAGHER D. Geometric modelling using Octree encoding. Computer Graphics and Image processing, vol 19, no 2, June 1982.

(21) PRATT M. Current research and development in solid modelling. Proceedings of the international conference 'The solid modelling revolution: are you ready?', Framingham, Massachusetts, October, 1984.

(22) PIPES A. Solid modellers: this year's style. CADCAM international, September, 1985, p 20-23.

(23) WICKENS L P. Presentation of Leeds University modelling system (BOXER). Proceedings CAM-I geometric modelling seminar, Cambridge, December 1983, p 113-135.

(24) GEISGOW A D. Results of benchmarks using Medusa. Proceedings CAM-I geometric modelling seminar, Cambridge, December, 1983.

(25) PARKINSON A. Geometric models - a database for computer aided NC data generation and process planning. Computer Aided Engineering Group, Cambridge University Engineering department, 1983.

(26) DASSLER R et al. Databases for geometric modelling and their application. Proceedings of IFIP, working conference, North Holland Publishing Co., Amsterdam, 1982.

(27) SPUR G, GERMER H J. Three dimensional solid modelling capabilities of the COMPAC system and some applications. Proceedings of ACM/CAE workshop in geometric modelling, Milan, February, 1982.

(28) BROWN C M, VOELCKER H B. The PADL-2 project. Proceedings 7th conference on Production Research and Technology, New York, September, 1979.

(29) BARNES R D. CAM-I automated process planning, the CAPP system. Proceedings CAM-I international seminar, Atlanta Georgia, April 1976, p 284-302.

(30) TULKOFF J. Use of the OPITZ classification code to implement the CAPP system. Proceedings CAM-I international seminar, Atlanta Georgia, April 1976, p 303-323.

(31) LINK C H. CAPP-CAM-I automated process planning system. Proceedings N/C conference, CAM-I inc, Arlington, Texas, 1976.

(32) MIPLAN. CV-MIPLAN. Computervision Corporation, 1981.

(33) MIPLAN. Group technology CAD/CAM. Organization for Industrial Research, Waltham, Massachusetts, 1983.

(34) EL-MIDNAY T T, DAVIES B J. AUTOCAP - a dialogue system for planning the sequence of operations for turning components. International Journal of Machine Tool Design Research, 1981, vol 21, no 3/4, p 175-191.

(35) ESKICIOGLU H, DAVIES B J. An interactive process planning system for prismatic parts (ICAPP). International Journal of Machine Tool Design Research, 1981, vol 21, no 3/4, p193-206.

(36) SSEMAKULA M E, DAVIES B J. Further development of the ICAPP system. UK-Hungarian seminar on computational geometry for CAD/CAM, Cambridge, England, 12-14 September, 1983.

(37) SSEMAKULA M E. Integrated process planning and NC programming for prismatic parts. Proceedings 1st International Machine Tool Conference, Birmingham, England, 26-28 June, 1984, p 143-.

(38) WATERBURY R. Computer aided process planning - key to cost savings. Assembly Engineering, June 1980.

(39) HEGLAND D E. Out in front with CAD/CAM at Lockheed-Georgia. Production Engineering, November 1981, p 45-48.

(40) C-PLAN. Computer aided process planning. CADCentre Ltd, Cambridge, England, 1984.

(41) CAPES. Computer aided planning and estimating system. Inbucon Productivity Services, Conventry, England, 1984.

(42) KOLOC J. MITURN a computer aided process planning system for numerically controlled lathes. Proceedings 2nd International Conference on Product Development and Manufacturing Technology, University of Strathclyde, 1971, p236-253.

(43) CROOKALL J R, SMITH J A. Program AUTOPLAN for the derivation of operations and machining conditions in multi-operation turning. Proceedings 2nd International Conference on Product Development and Manufacturing Technology, University of Strathclyde, 1971, p 308-323.

(44) SPUR G, ANGER H M, KUNZENDORF W, STUCKMANN. CAPSY - a dialogue system for computer aided manufacturing process planning. Machine Tool Design and Research Conference, 1978, vol 19, p 281-288.

(45) HANNAM R G, PLUMMER J C. Capturing production engineering practice with a CADCAM system. International Journal of Production Research, 1984, vol 22, no 2, p 267-290.

(46) PLUMMER J C, HANNAM R G. Programing towards a flexible unmaned factory using a CADCAM system. Proceedings of the Second European Users Conference, Brussels, 1983.

(47) DAVIES B J, DARBYSHIRE I L. The use of expert systems in process planning. CIRP Annals, vol 33/1, 1984, p 3Ø3-3Ø6.

(48) DARBYSHIRE I L, DAVIES B J. EXCAP: An expert generative process planning system. ACME Report on workshop on manufacturing engineering, Nottingham University, 26-28 September 1984.

(49) LEWIS W C, BARTLET E, FINFTER I I, BARASH M M. Tool oriented process planning. Proceedings 23rd International Machine Tool Design and Research Conference, 1982, p 421-427.

(5Ø) CHANG T C, WYSK R A. An integrated CAD/automated process planning system. AIIE Transactions, September 1983, vol 13, p 223-232.

(51) CHANG T C, WYSK R A. Integrating CAD and CAM through automated process planning. International Journal of Production Research, 1984, vol 22, no 5, p 877-894.

(52) CHANG T C, WYSK R A. An introduction to automated process planning systems. Prentice-Hall, 1985, p 186-216.

(53) PURDUE J E, RICHARDS J E, DAVIS R S. PROPLAN The PERACAM automatic process planning system. PERA, Melton Mowbray, Leicestershire, Report no 376, June 1983.

(54) LOGAN F A. The five stages to automated process planning. The American Society of Manufacturing Engineers Conference, Washington DC, USA, 14-15 March 1984.

(55) LOGAN F A. Process plans via manufacturing codes. Proceedings 7th Annual Conference on Design Engineering, Birmingham, England, September 1984.

(56) LOGAN F A. Process planning - the vital link between design and production. Proceedings 6th Annual Conference on Design Engineering, Birmingham, England, 4-6 October, p 37-.

(57) ROUND D. Report on XPS-1. Department of Engineering Production, Loughborough University of Technology, Loughborough, Leicestershire.

(58) SACK C F. Machine Data Representation in XPS-1 and specification for XPS-2. Final report presentation, CAM-I contracy C-84-PP-Ø3.

(59) STEUDEL H J. Computer-aided process planning: past, present and future. International Journal of Production Research, 1984, vol 22, no 2, p 253-266.

(6Ø) CHOI B K, BARASH M M, ANDERSON D C. Automatic recognition of machined surfaces from a 3D solid model. Computer-aided design, March 1984, vol 16, no 2, p 81-86.

(61) MILL F, SPRAGGETT S. Artificial intelligence for production planning. Computer-Aided Engineering Journal, December 1984, p 21Ø-213.

(62) EVERSHEIM W. Automatic generation of process plans and NC tapes for sheet metal parts. The Production Engineer, March 1981, p 28-31.

(63) EGELIE K A. AUTOPROS - automated process plannning system: problem definitions, project history and conclusions. Proceedings CAM-I International CAM Seminar, Bournemouth, England, 2 May, 1975, p 182-195.

(64) ZHANG S, GAO W D. TOJICAP a system of computer aided process planning for rotational parts. Annals of the CIRP, 1984, vol 33, no 1, p 299-301.

(65) CHOI B K, BARASH M M. STOPP: an approach to CADCAM integration. Computer-aided design, vol 17, no 4, May 1985, p 162-168.

(66) MICHIE D. Machine intelligence and related topics. Gordon and Breach, 1982

(67) MICHIE D. Expert systems. The computer journal, vol 23, no 4, 1980, p 369-376.

(68) PARKINSON A. Expert systems in geometric modelling and its application. Computer Aided Engineering Group, Cambridge University Engineering Department, Cambridge.

(69) BARR A, FEIGENBAUM E. The handbook of artificial intelligence - Volume 1. Pitman, 1981.

(70) BARR A, FEIGENBAUM E. The handbook of artificial intelligence - Volume 2. Pitman, 1982.

(71) SHORTCLIFFE E A. Computer-based medical consultation: 'MYCIN'. Elsevier, 1976.

(72) SANTA T, KIMURA F, SUZUKI H, FUJITA T. Designing machine assembly structure using geometric constraints in product modelling. Annals of the CIRP, vol 34, no 1, 1985, p 169-172.

(73) HART N, CASE K, BENNATON J, ACAR S. A CAD engineerring language to aid manufacture. Proceedings of the International Conference on Computer-aided Production Engineering, Edinburgh, 2-4 April, 1986.

(74) WEIZENBAUM J. Computer power and human reason: from judgement to calculation. Penguin, 1984, p 182-201.

(75) CHARNIAK E, McDERMONT O. Introduction to artificial intelligence. Addison-Wesley, 1985, p 169-246.

(76) GILT. Command processor user guide. The CAD Centre, Madingley Road, Cambridge, January, 1981.

(77) ISA B S, EVEY J, McVEY B W, NEAL A S. An empirical comparison of two metalanguages. International Journal of Man-Machine Studies, 1985, vol 23, p 215-229.

(78) BENNATON J, CASE K, HART N, ACAR S. A system to aid design by planning manufacturing operations. Proceedings of 1st National Conference on Production Research, University of Nottingham, 9-11 September 1985.

(79) ACAR S, CASE K, BANNATON J, HART N. Evaluaton of a new approach to design and process planning. ACME Report on workshop on manufacturing engineering, Nottingham University, 26-28 September 1984.

(80) NEWSTED P R. Paper versus online presentations of subjective questionnaires. International Journal of Man-Machine Studies, 1985, vol 23, p 231-247.

(81) WICKENS L P. Dimensions and tolerances in NONAME. Department of Mechanical Engineering, University of Leeds, Leeds, June 1986.

(82) HART N. The familiar engineering language program and development utilities. Department of Engineering Production, Loughborough University of Technology, Loughborough, August 1986.