Aston University

Some pages of this thesis may have been removed for copyright restrictions.

If you have discovered material in Aston Research Explorer which is unlawful e.g. breaches copyright, (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please read our Takedown policy and contact the service immediately (openaccess@aston.ac.uk)

INTEGRATING COMPUTER AIDED ENGINEERING FUNCTIONS: THE MANAGEMENT OF INFORMATION

Emilius Mwape Pensulo

Doctor of Philosophy

THE UNIVERSITY OF ASTON IN BIRMINGHAM

March 1987

This copy of the thesis has been supplied on condition that anyone who consults it is understood to recognise that its copyright rests with its author and that no quotation from the thesis and no information derived from it may be published without the author's prior, written consent.

THE UNIVERSITY OF ASTON IN BIRMINGHAM

Title: **Integrating computer aided engineering functions: the management of information**

Name: **Emilius Mwape Pensulo**

Degree: **Doctor of Philosophy**

Year: **1987**

SUMMARY

The integration of computer aided engineering functions is one of the most pressing requirements in the industrial computing field today, as evidenced by the interest shown in the concept of computer integrated manufacturing (CIM). However, major obstacles exist at many levels. Within individual enterprises, the main problem seems to be the inability of discrete software systems to share data effectively. This was the problem addressed in this research work. The solution was seen to lie in a full analysis and centralized management of the common data in the system.

There are three main sections in this thesis. Firstly, the formulation of a generalized methodology for the realization of a discrete mechanical product is outlined in order to expose the domain for integration. Specific proposals are then presented regarding the design of integrated CAE systems based on the database approach to information management. These proposals arose from a study of the specific nature of engineering data and activities and also from those of discrete mechanical products. The main problem identified was the maintenance of data integrity in an engineering database: this led to the two main proposals made in the thesis i.e. the introduction of deductive capacities in the conventional database system architecture and the layering of the database structure into a global and several local components. The last section of the thesis concerns the design, implementation and evaluation of an integrated CAE system, created in order to enable an objective evaluation of the proposals arising from the research to be carried out. The specific application chosen was the design of industrial gearboxes.

It is now generally recognized that integration has to involve some form of centralized database management. The subject of this thesis is therefore likely to remain of interest for sometime to come.

Keywords/phrases: Computer aided engineering systems, Engineering databases, Integration, Engineering data modelling

To Joyce, Chibesa and Mwila for giving me all the time I needed.

A C K N O W L E D G E M E N T S

In the course of the research work on which this thesis is based, I received much goodwill and assistance from many people only some of whom I can mention here.

Firstly, I would like to thank my supervisor, Mr. T H E Richards, very sincerely for the fruitful guidance and unfailing support accorded to me throughout the period of the research. Without the benefit of his experience and broad approach to the research topic, it would have been extremely difficult to acquire a proper perception of the wide field in which I was working, let alone carry out a sufficiently complete piece of research work in it. I also greatly valued the trust which underlined our working relationship.

I would also like to record my sincere gratitude to many members of staff, in particular Mr. G A Jones for his assistance in matters to do with computer interfacing and Mr. J Scattergood for his excellent cooperation with regard to the purchasing and maintenance of computer hardware and software.

Last but not least, I wish to acknowledge the valuable knowlegde gained from several contemporaries, especially K. Butler in the general area of microcomputing and M. Bava and S Williamson-Taylor working in the related field of artificial intelligence-based expert systems.

TABLE OF CONTENTS

CHAPTER THREE

DEVELOPMENT OF AN INFORMATION TRANSFORMATION MODEL FOR THE DISCRETE-PRODUCT REALIZATION PROCESS

CHAPTER FOUR

INTEGRATED COMPUTER AIDED ENGINEERING SYSTEMS

CHAPTER FIVE

THE DATABASE APPROACH TO INFORMATION MANAGEMENT

CHAPTER SIX

APPLICATION OF THE DATABASE CONCEPT TO THE MANAGEMENT OF INFORMATION IN COMPUTER AIDED ENGINEERING SYSTEMS

CHAPTER EIGHT

EVALUATION OF THE INTEGRATED COMPUTER AIDED ENGINEERING PACKAGE

CHAPTER NINE

GENERAL DISCUSSION AND CONCLUSIONS

APPENDICES

LIST OF FIGURES

C H A P T E R   O N E

OBJECTIVES, METHOD OF INVESTIGATION AND MAIN CONCLUSIONS

1.1 Introduction

The application of computer technology to engineering functions has accelerated tremendously over the last ten years. Despite the proliferation of CAE systems however, recent surveys have revealed disappointment by a significant number of manufacturing companies with the returns on their CAE investments [**1**]. This is primarily because, in general, CAE is still a fragmented technology whereby computers are used to aid either distinct or closely-related engineering functions with little attention paid to the need for integration with other functions in the organization. Such piecemeal application of computer technology has meant that the many potential benefits of CAE have, in many cases, not been realized.

1.2 Justification for the research work

The lack of integration in most current CAE systems can be attributed to a variety of reasons. The main one however seems to be the incompatibility of discrete systems with one another, with regard to their data management features. Users have become increasingly aware of the enormous value of the data built up on individual CAE systems for a variety of other functions in the organization. Hand in hand with this awareness has been the demand for more and better ways of getting at that data and associating it with other computer-held information, perhaps in a way not originally envisaged. It was in view of this demand that the research work on which this thesis is based was undertaken.

1.3 Objectives of the research work

The research was focussed on the critical issue of data organization in a manufacturing enterprise with a view to facilitating integration. The main objective was to formulate a methodology for organizing information in a computer-aided engineering environment to support the discrete-product realization process for a mechanical product. In view of the large scope for integration in engineering systems, it was decided to concentrate the research on the integration of mechanical engineering design systems while taking the necessary care to ensure that the ideas developed could be extended to other fields.

1.4 Main phases of the research work

To pursue the objective stated, three main phases were identified for the research work as follows:-

(1) The development of an information transformation model for the discrete-product realization process. This phase served (a) to determine the information transformation requirements at the various stages of the realization process and (b) to reveal special characteristics of engineering information which needed to be taken into account when formulating the CAE information processing model.

(2) The development of a computer-based information processing model applicable to the transformation model developed in the first phase.

(3) Implementation of the CAE system designed. In the end, only a partial system could be implemented but this was complete enough to demonstrate the feasibility of the proposed approach to the management of information in CAE systems.

1.5 **Main results and conclusions**

The main results and conclusions arising from the research can be divided into two groups i.e. general ones relating to the strategy for integration and those specifically about the database approach proposed as a route to integration. These are summarized below.

(a) The general strategy for integration

The research led to the identification of two major requirements. These are (a) an adequate analysis of the functions targeted for integration and the data associated with them and (b) the mapping out of a strategy for the implementation of the integrated system. Both these requirements point to a necessity for a centralized approach to the problem of integration.

(b) The database approach to information management in CAE systems.

This is shown in this thesis to be a viable route to integration. Its main attraction is the shareability of data that it facilitates. Some conclusions could be reached with regard to two main concerns as follows:-

- Applicability of conventional database concepts and technology.

The package created for gearbox design is based on an established data model (relational) and involves the use of a proprietory database management system (DBASEII). The adequate performance given by the system seemed to confirm the applicability of these conventional database tools to the problem of CAE integration even though some novel features i.e. deductive capabilities and a multi-layered database structure, had to be introduced in the conventional database system architecture in order to maintain the integrity of the data in the database.

- Analysis and modelling of engineering data for integration.

Two main conclusions were drawn regarding the modelling of engineering data. These are (a) that only truly common data i.e. data required as input to other applications in the engineering system should be centrally managed and (b) that among the main conventional data models, the relational model is the one that offers the level of flexibility required in integrated systems.

Apart from the more specific results and conclusions outlined in the preceding paragraphs, the research also led to an important general conclusion. This is that, because of the multiplicity of user requirements, a realistic integrated CAE system should necessarily be a quasi-expert system where the knowledge base is created within the user organization. This puts the onus for integration on the user: the least that should be expected of suppliers of software suites is full documentation of and access to the data required and created by their systems.

The main problem identified was the maintenance of integrity of the data on the database of an integrated system, in view of both the high sensitivity of engineering data to change and the iterative nature of the discrete-product realization process. This problem warrants further research.

CHAPTER TWO

GENERAL OUTLINE OF THE RESEARCH WORK

2.1 Introduction

A brief description of the objectives, the strategy and the main results of the research undertaken has been presented in the preceding chapter. The main objectives in this chapter are to (a) amplify on the goals of the research work and (b) establish the rationale for the method of investigation adopted. Accordingly, the working notion of an integrated computer aided engineering system is first presented, followed by an extended description of the three main phases of the research work enumerated in Chapter 1. Lastly, brief introductory notes are given about each of the subsequent chapters in the thesis.

2.2 Integration in computer aided engineering systems

Computer aided engineering can be implemented in an uncoordinated manner whereby several standalone computer hardware and software systems are used to aid individual activities or departments within an enterprise. Of late however, there has been a widespread realization of the many benefits that can be obtained from the integration of these systems: this is evidenced by the growth in popularity of the concept of computer integrated manufacturing (CIM). In this research, an integrated CAE system is defined as one in which the computer-based tools are linked together, the objective being to optimize the operation of the total engineering system over and above that of the individual components.

There are currently a large number of factors hindering the implementation of integrated CAE systems. Some of these are at the level of the computer industry as a whole while others arise within enterprises. At the company level, the main technical problem is usually the lack of compatibility among the discrete CAE elements that may exist to aid the different aspects of the company's operation. The incompatibility can be of two kinds i.e. (a) the inability to transfer data from one system to another and (b) the inability of two or more software elements on the same computer system to share data effectively. The former is basically a communication problem which is likely to become less significant in the near future with the development of standards such as MAP (manufacturing automation protocol). The latter however can, in general, not be solved by such developments as it involves the understanding and modelling of the data to be shared. This was the problem addressed in this research work.

The major concern in the research was therefore the optimum management of the information in a computer aided engineering system so as to facilitate the integration of the various software elements in it. The particular problem posed was the integration of the various applications involved in the development process for a discrete mechanical product.

2.3 The method of investigation adopted

The research was seen to require three distinct phases as follows:-

(1) The development of an information transformation model for the discrete-product realization process.

(2) The development of a computer-based processing model applicable to the transformation model developed in the first phase.

(3) The implementation of a specific integrated CAE system based on the models developed in the first two phases.

In addition, it was recognized that in order to ensure the general applicability of any integration methodology eventually developed, it was necessary that the first two phases were carried out at a generalized level. The phases listed are discussed individually in the subsections that follow.

2.3.1 Development of a generalized methodology for the realization of discrete-products.

The development a new product can generally be viewed to start from when a need is recognized in the market place and end when a product to satisfy the need has been manufactured and released for sale. In this thesis, the term 'product realization process' has been used frequently to refer to the whole range of activities in this process.

A working party of the Institution of Production Engineers **[2]** made the important observation that the general principles underlying the product realization process in all the manufacturing firms included in their survey appeared to be the same, regardless of the type of product, the volume of production or the size of the firm. In the research work, it was precisely the integration of these stages that was being sought: it was therefore necessary that these were described in more exact terms in the first place.

The various stages in the development process for any product are linked together mainly through their usage of common data about both the product itself and all the resources required in the process. It was therefore recognized at an early stage of the research work that any integration methodology eventually developed would involve the optimum management of the data in the manufacturing system. Therefore, after identifying the various stages, a necessary task was a study of the information transformation requirements at each of the stages.

2.3.2 Development of a computer-based information processing model

The good coordination of the various stages of the realization process for a product is a necessity in every type of manufacturing enterprise, including those in which computer technology has not been applied. In CAE systems however, the increased need for productivity in today's manufacturing industry requires that the coordination of the various computer aided manufacturing elements is itself computer-based. This is the essential idea in computer integrated systems of every kind.

Since the linkage of the various stages of the product realization process is through their usage of common data, the centralized modelling of the common data in the CAE system was recognized at this stage as a viable route to their integration. This is basically the database approach to information management. In the end, the basic task posed at this stage was the development of a methodology for the design and implementation of integrated CAE database systems.

2.3.3 **Design and implementation of an integrated CAE system**

The first and second phases of the research were necessarily theoretically-based and led to generalized proposals regarding integrated CAE systems. It was therefore found necessary to design and implement an integrated system in order to demonstrate and allow the evaluation of the proposals made in the first two phases.

The CAE package implemented is for a specialized application: it is yet not possible to create generalized integrated systems. For a number of pragmatic reasons, the application chosen was the design of industrial gearboxes involving spur or helical gears only. The package is based on database concepts and involves the use of a proprietory database management system. Despite its limited scope, the package seemed to have served the purposes of the research work and also has the potential to be developed into a more elaborate CAE system.

2.4 **Outline of the rest of the research work**

Following the brief summary of the thesis given in the first chapter and the extended discussion of the objectives and method of investigation given in the current one, the rest of the thesis has been arranged as follows:-

Chapter 3: Development of an information transformation model for the discrete-product realization process

Chapter 4: Integrated computer aided engineering systems

Chapter 5: The database approach to information management

Chapter 6: Application of the database approach to the management of information in computer aided engineering systems

Chapter 7: An integrated CAE system for the design of industrial gearboxes

Chapter 8: Evaluation of the computer aided engineering package

Chapter 9: General discussion and conclusions of the thesis

Brief introductory notes about each of the chapters are given in the subsections that follow, in order to indicate the scope of the research work and also establish the flow in the thesis.

2.4.1 Development of an information transformation model for the discrete-product realization process (Chapter 3)

The work done in the first phase of the research is presented in this chapter. The formulation of a general methodology for the development of a product is presented, followed by an analysis of its information transformation requirements. The methodology proposed ranges from the recognition of a need, through design and manufacture, to the marketing of a product to fulfill the need. This chapter establishes the domain for integration work in CAE systems.

2.4.2 Integrated computer aided engineering systems (Chapter 4)

The current application of computers in engineering is reviewed firstly in this chapter. The working notion of an integrated CAE system is then introduced, followed by an extended discussion of the factors that currently hinder their implementation. The discussions in the chapter led to the recognition of a centralized management of information as a viable route to integration.

2.4.3 **The database approach to information management (Chapter 5)**

The identification of centralized information management as a feasible approach to integration led to a study of the database approach which has so far been successfully applied to the management of commercial information. The main features of the approach and the current practices and technology associated with it, such as data models, database design methodologies and database management systems are reviewed at length in this chapter in order to justify the consideration of the approach for application in CAE systems.

2.4.4 **Application of the database approach to the management of information in computer aided engineering systems (Chapter 6)**

Specific proposals regarding the design of integrated CAE systems based on the database approach are presented in this chapter. These were drawn up after a detailed study of the special characteristics of engineering data and applications and also of discrete mechanical products. The main proposals are the introduction of limited deductive capabilities into the normal database system configuration and the layering of the database structure into a global and a set of local components.

2.4.5 An integrated CAE system for the design of industrial gearboxes (Chapter 7)

The design and implementation of a package for gearbox design, created in the course of the research work is described in the chapter. The presentation is centred around the design of a control module (which provides the deductive capabilities as proposed) and that of the database structure. Extended descriptions of the database management system used, the various application programs, the contents of the conceptual database and the design of a data directory/dictionary module for the system are presented in Appendices 1 to 4 respectively.

2.4.6 Evaluation of the integrated computer aided engineering system (Chapter 8)

The package implemented is evaluated in this chapter. After justifying its designation as an integrated system, the compliance of the system with the design specifications enumerated in Chapter 7 is assessed, followed by an examination of the general applicability of the procedures used in its design and implementation. The package was assessed to have served the purposes of the research work.

2.4.7 **General discussion and conclusions of the thesis**

The discussions and the main conclusions presented are in two categories i.e. those regarding the general requirements recognized for integration and those specifically about the database approach proposed. The main conclusions reached have been mentioned in brief in Chapter 1. Specific proposals regarding future research in integrated CAE systems have also been given in the chapter.

2.5 **General remarks**

The research was carried out over the three-year period from 1983 to 1986. This period witnessed a virtual explosion in the application of computer technology in engineering, leading to a greater prominence of the idea of integration. With regard to the use of databases in engineering systems, the literature reflects a shift over the period, from the idea of one centralized company database initially promoted to the more distributed systems enabled by developments in communication standards. The bulk of the research work undertaken focussed on the the centralized modelling of the data in any CAE system: this was recognized as a primary requirement, regardless of whether the system is to be installed on one computer or on a network. The application of databases in engineering is therefore likely to remain an important research topic for some time.

C H A P T E R   T H R E E

DEVELOPMENT OF AN INFORMATION TRANSFORMATION MODEL FOR THE DISCRETE-PRODUCT REALIZATION PROCESS

3.1 Introduction

In the broadest sense, a product may be defined as an article or a commodity which results from a process [**3**]. In the case of a discrete product, this process consists of four main phases i.e. the creative design of a functional system, the development of the most effective design consistent with production economics, the strategic planning to coordinate production and sales factors and lastly, the sustained production of hardware. The bulk of the work presented in this thesis is to do with the optimization of the information processing function during the production realization process in a computer aided engineering enviroment. At the start of the project, it was recognised that the existence of a coherent and disciplined approach to the task of realizing a discrete product was a prerequisite for any meaningful study of information processing issues within such a process. It was therefore found necessary to establish a product realization methodology and thereafter, an information transformation model on which to base the study of information processing issues.

The development of an information transformation model for a general product realization process is outlined in this chapter. The desirability of general methodology is first established, followed by a literature review of product evolution methodology formulations. A specification for and an outline of a proposed general methodology is then presented, followed by an analysis of the information flow requirements implied by the proposed methodology. This analysis revealed a wide range of information transformation requirements all of which were to be considered during subsequent work.

3.2 The feasibility of prescribing a general methodology for discrete-product realization

The importance of the product evolution process and, therefore, the need for a rationalised approach to carrying out the process are generally well acknowledged. Beyond this however, the prescription of a generalised methodology for the process, applicable in different types of firms and to different types of products, has generally been regarded as an impracticable task. Many firms have therefore adopted an ad-hoc approach to the product evolution process, only conforming to some codes of practice in the relatively well-studied areas such as element design, drafting and hardware manufacturing.

There is now growing recognition of the existence of some common product evolution principles. A working party of the Institution of Production Engineers [**2**], after studying working practices in a number of different manufacturing firms, made the important observation that whilst product evolution procedures used by many firms were obviously tailored to suit specific industries, the general principles underlying these procedures appeared common, irrespective of the size of firm, the type of product or the volume of production. Also, the proliferation of proprietary computer software and hardware aimed at facilitating integrated manufacturing can, to some extent, be taken to signify recognition of the existence of common product evolution principles. It therefore seems that the proposition of a general methodology for product realization that embraces these principles is indeed a feasible task.

3.3 Benefits envisaged from the use of a generalised product realization methodology

Currently, procedures with respect to the earlier stages of the product realization process, such as conceptual and functional design, are not as well prescribed as those for the later stages, such as detail design and production. Notwithstanding the current situation, a methodology linking all the phases from the recognition of a need to the production and marketing of a product would provide the following important benefits.

(a) Allow a rational and substantiated stepwise progression of work, thus avoiding dead-ends, duplication of work, delays and expensive corrective action at later stages.

The methodology would remove the constant problem of deciding what needs to be done next, leaving the management and staff free to concentrate on the product being created.

(b) Facilitate efficient allocation of functions and responsibilities relating to the product evolution process to departments and individuals in an enterprise.

(c) Enable easier monitoring of the whole process and therefore, more informed and timely decisions by management at critical stages of the process.

(d) Enable optimal structuring of the data resource of the enterprise to support the product,the design and realization process more efficiently.

Nowadays, computers are being used increasingly to aid various activities in the product creation process. According to **Eberlein** and **Wedekind** [**4**], the development of the next generation's integrated computer aided engineering (CAE) systems depends critically on the existence of a general methodology for product design. Since such a methodology can only be part of a generalized one covering all the activities in the product realization process, the above statement is even more relevant to the latter. This explains why it was decided to precede the study of information processing in computer aided systems by the establishment of a general methodology for the product creation.

3.4 **Literature survey on a general methodology for**

product realization

While the importance of the product realization process is well recognised, its presentation as one inherently continuous process is rare in current literature. Instead, detailed prescriptions of aspects or phases of the process can be found in literature belonging to traditionally different disciplines within the manufacturing field. The main disciplines concerned with product evolution are (a) Engineering design (b) Engineering production and (c) Management. A brief review of methodology-related literature within each of these main disciplines is given in the sections that follow.

3.4.1 Engineering Design

Engineering design has been defined as the process of transforming information from a customer's statement of requirements to a full description of the proposed technical system to satisfy the requirements [**5**,**6**]. It is a hybrid activity which depends upon a proper blending of art, science and mathematics for its successful execution. realizing the importance of the process, a number of workers in the field, notably **Matousek** [7], **Dixon** [8], **Jones** [9], **Woodson** [6], **Pahl** & **Beitz** [**10**] and **Hubka** [5] have proposed systematic approaches to the process. All these proposals contain the following important modules:-

(a) Goal recognition i.e validation of the market need and identification of the type of solution required.

(b) Task specification i.e. description of a more specific task to be done which will accomplish the goal.

(c) Design synthesis and analysis i.e. generation and analysis of a number of candidate solutions to the problem and the selection of the optimum solution.

(d) Solution specification i.e translation of the selected solution into production terms.

The methodology proposed in the referenced literature, commonly referred to as the 'Design Method', is iterative and the main concerns during the process are the functionality, economic feasibility, ease of production, ergonomics and aesthetics of the product under development.

3.4.2 **Engineering production**

With regard to the product evolution process, production engineering literature highlights two major phases within the field. They are (a) Process engineering and (b) Prototype manufacture.

(a) In process engineering, the information created in the design phase is used to plan for the manufacture of the product. A two-stage procedure is commonly propounded [**11,12,13,14**] as follows:-

- Production planning. Typically, steps or operations for each process are listed on process sheets together with the machines, equipment, tools and materials required. Expected performance standards are also specified. The purchasing of all bought-out materials and products is also planned.

- Preparation for production. Typically, production machinery are purchased and installed, personnel are hired or trained as necessary, full production is scheduled and production consumables are purchased.

(b) Product manufacture refers to the steady state phase when all the production elements planned are brought together to sustain production of the items as dictated by the market.

Both these phases require efficient management of manpower, money, machinery, materials and information. Accordingly, some production engineering literature give an extended coverage to issues traditionally associated with the management field, such as costing, purchasing, stock control and production economics. [**15,16**].

3.4.3 **Management**

The main management functions i.e. planning, direction and control are relevant to every phase of the product evolution process; indeed the process is normally initiated by a management decision. Consequently, management literature on the product realization methodology tends to cover the whole process, with varying degrees of generalisation and with the expected emphasis on less quantitative and non-engineering issues.

Carson and **Rickards** [17] present the process as consisting of three major phases i.e. opportunity search, product development (engineering) and commercialisation. **Douglas, Kemp** and **Cook** [18] describe a systematic procedure to new product development consisting of five main stages i.e. analysis of an enterprise's existing strength, identification of the markets to which these strengths can be fitted, idea generation, brand building (i.e the engineering and testing of a product) and lastly the forecasting of potential sales in relation to the enterprise's profit objectives.

The product evolution process as seen generally from the management viewpoint is briefly summarised by **Crawford** [19] under the following headings:-

(a) Exploration i.e. the search for product ideas to meet company objectives.

(b) Screening i.e analysis to determine which ideas are pertinent and merit more detailed study.

(c) Business analysis i.e. the expansion of the idea, through creative analysis into a concrete business recommendation including product features and a production program

(d) Development i.e turning of the idea on paper into a product that can be demonstrated and reproduced.

(e) Testing i.e. carrying out of any commercial experiments necessary to verify earlier business judgements.

(f) Commercialisation i.e. the launching of the product in full-scale production and sale.

A number of other workers [**20,21,22**] present the process along similar lines, under the general theme of new product management.

3.4.4 **Interdisciplinary formulations**

A detailed description of the acvities in the product realization process is given in a 1975 publication of the Institution of Production Engineers [**2**]. Published as a 'code of practice' to assist managers in achieving the most effective evolution of their product, the formulation divides the evolution process into four main stages as follows:-

(a) Product initiation i.e. the identification of a requirement for the product on the market and consideration of the technical and economic feasibility of the product.

(b) Design and development i.e. the assessment of the proposed new design in detail, usually involving the manufacture and testing of a prototype.

(c) Pre-production i.e. ensuring that the product will be acceptable to the market when introduced and that it can be produced within time and cost targets.

(d) Production i.e. full production of hardware as dictated by the market conditions.

Apart from specifying the various activities within each of the four stages, the code also outlines the functions of individual departments, the company management and the directors at critical stages in the process.

Less detailed interdisciplinary formulations of a general methodology for product realization are have also been carried out by **Bolz** [**13**], **Kazanas** [**14**] and **Richards** [**22**].

3.5 Formulation of a general methodology for discrete-product realization

The main interest during the literature survey was in the flow of information during the product realization process. An important outcome from the survey was the realization that most of the formulations found in the literature prescribed the various activities in the process with no particular attention paid to the flow of information during the process. It was therefore found necessary to establish a methodology focussed on information flow so as to facilitate the subsequent study of information processing issues. The formulation of such a methodology is outlined in this section.

3.5.1 Specifications for a general methodology

The potential usefulness of a general product realization methodology has been discussed earlier in this chapter (Sect 3.3). In order to yield the benefits expected, it was considered that the general methodology should provide the following important features:-

(a) A framework for the whole of the product realization process and not just cover only those activities belonging to a particular discipline, such as engineering.

(b) Comprehensive definitions of activities to be performed. Those activities which can be reduced to a formal description should be specified in a sufficiently detailed form as to provide a logical framework for implementation. The definitions should distinguish between routine procedures and those involving critical decision-making.

(c) Ability to check for completeness and quality of work at critical stages of the process.

(d) Ability to cater for a wide range of product development environments i.e. it should be possible to construct a product-specific plan by restricting the parameters of the general methodology.

(e) Flexibility in providing guidance without restricting creativity or preventing the adoption of new techniques and new technology as these become available.

(f) Recognition that not all product parameters can be specified in the initial stages of the development process. The methodology should therefore allow iteration as ideas develop or change.

To enable the study of information processing during the product realization process, an important additional specification imposed was that the methodology provided guidelines on information transformation requirements at each of its phases. The end result would then constitute an information transformation model for the product evolution process.

3.5.2 **Modularisation of the product realization process**

The product realization process is basically a continuous one. It involves the repeated specification of tasks, application of the necessary resources, including the information resource, to the set tasks, evaluation of the results obtained and decision-making as to the next task. For many practical purposes, including efficient implementation, it is necessary to reduce the complexity of the process by dividing it into a number of convenient modules. The process can therefore be viewed to constitut an ordered set of activities

$$A = (A_1, A_2, A_3, \ldots\ldots, A_k)$$

where (a) each module Ai has a set of inputs I_{Ai}, a set of outputs O_{Ai} and a method, whether direct or heuristic for obtaining the output set from the input set, and

(b) a subset of the output set O_{Ai-1} forms an essential subset of the input set I_{Ai}.

The terms 'input' and 'output' above are used in a very general sense as, in practice, these can take many forms such as drawings, analysis figures and hardware.

The number of modules used to describe the product development process can vary, depending on the viewpoint and purpose as can be deduced from the literature review given in Sect 3.4. On the other hand, the notion that there exists a set of common product evolution principles [**2**] suggests the feasibility of optimising the number of modules in a general methodology.

3.5.3 **Iteration during the product realization process**

While viewing the product evolution process as a sequential one, it is recognised that in practice, the process involves iteration at two levels as follows:-

(a) Process level

If the output from an activity A_i is considered unsatisfactory, then either the activity has to be repeated or the whole process is reverted to an appropriate earlier stage A_m, $(m<i)$.

(b) Module level

In general, iteration will occur within each module to successively improve the quality of the output even before the output is subjected to evaluation at process level

In practice, the sequential and yet iterative nature of the product evolution process results in some overlapping among the various activity modules. The modularisation approach therefore only indicates the succession of important events in the process and does not imply a strict chronological order.

3.5.4 Outline of the proposed general methodology for discrete product realization

The proposed general methodology for discrete-product evolution is shown in flowchart form in Fig 3.1. The proposal encompasses the ideas and general principles propounded in the literature, most of which is cited in Sect 3.4. An activity shown within a rectangle moves the process forwards while one shown with a diamond is a check for completeness and quality at a critical stage of the process. A brief description of the objective(s) of each of the twelve modules and the logical connections among them is given below.

(1) Recognition of a need for a product

The stimulus to initiate a product may come from within or from outside an enterprise. Internal stimuli may include:-

- Spare production capacity
- A continuing drop in profitability
- New discoveries by the R & D department.

External stimuli may include:

- Technical or economic obsolescence of the present product(s).
- Availability of new data or technology through research.
- New consumer or legislative requirements.
- Technical and economic superiority of competing products.

Guided by the corporate objectives, a study of the market and the enterprise's resources will define the area in which the search for a new product can be usefully pursued. The functional requirements of the product being

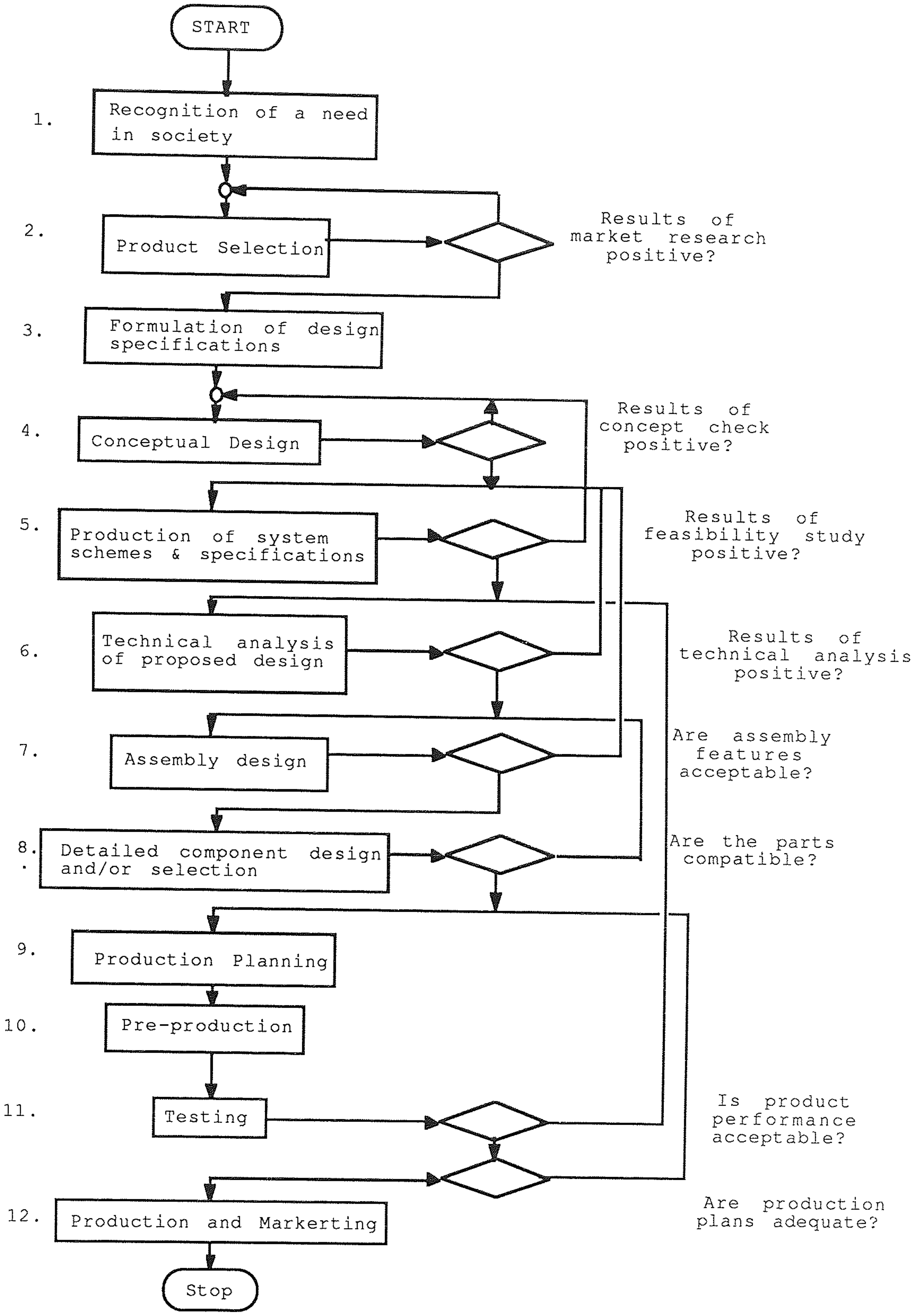


Fig 3.1 Showing a general methodology for discrete-product realization

sought can also then be identified and described in a very general form at this stage.

(2) **Product Selection**

Having identified both the 'search field' and the functional requirements for the product, the objective at this stage is to select those product ideas that seem to fit in best with the enterprise's objectives. The major steps in the selection process are:-

- Discovery of product ideas.
- Comparison of candidate products based on a number of technical and economic criteria and the selection of the optimum product.
- Specification of the most important features and requirements of the selected product. This product proposal is usually submitted to the company's board for approval before being acted upon.

The acceptability of the selected product should be confirmed by a research of the intended market.

(3) **Formulation of design specifications**

The objective at this stage is to specify the required performance of the selected type of product in order to enable its design. The required function(s), the appropriate inputs and outputs and the environmental conditions under which the required performance must be obtained are prescribed in quantitative terms where possible.

Apart from the required performance, other considerations during design specifications should include the state of technology, established industrial practice and the possible future developments in the particular field.

Once the product has been adequately specified and all parties concerned are satisfied that the listed requirements are technically and economically attainable, the way is clear for the conceptual design phase.

(4) **Conceptual Design**

Having properly established the design requirements in relation to the commercial and technical possibilities, it is then necessary to evolve a concept on which to base the design of the product. This refers to the establishment of design principles considered most suitable to meet the requirements and a statement of the form considered most suitable for the proposed product.

Generally, there will be more than one candidate concept for any design, each with its own strengths and weaknesses. The final concept adopted therefore represents a compromise.

Conceptual design constitutes the first cycle from the functional to the anatomical structure of the product. It is therefore essential to carry out a subsequent 'concept check' of the selected solution in order to confirm its feasibility and justify further expenditure of time and money in the later stages of the work.

(5) **Production of System Schemes and Specifications**

With the concept of the design established, the objective at this stage is to develop the design to the point where subsequent detail design can lead directly to production. The preliminary forms of the main components of the product are produced iteratively, using appropriate starting dimensions, specified loading and an embodiment of the concept into a rough anatomical arrangement of the product.

After establishing the systems schemes and specifications, it is possible to carry out a more quantitative feasibility study, especially with regard to the economics of the product. If the results of the study are unsatisfactory, the schemes and specifications must be revised or a different concept for the design may have to be sought.

(6) **Design Analysis**

In this second phase of embodiment design, the objective is to clarify and concretize further the preliminary system schemes and specifications. Detailed layouts of the designs are developed in the light of the results of detailed calculations of the static and dynamic characteristics of the system and components, or those obtained from prototype testing where necessary. Other considerations at this stage include the use of standard codes of practice, and the compatibility of the product with other product systems where relevant.

Embodiment design (i.e. modules (5) & (6)) involves a large number of corrective steps in which analysis and synthesis constantly alternate and complement each other. The search for solutions and their evaluation is complemented by the identification of design faults and by optimization. The combined use of information on analysis methods, manufacturing processes, materials, standard parts and practices etc., involves great efforts on the part of the designer at this stage.

(7) **Assembly Design**

So far in the product evolution process, only the functional and economic features have been considered in great detail. The objective at this stage is to transform the functional schemes at hand into a feasible assembly. Apart from ensuring dimensional compatibility, it becomes necessary at this stage to consider in greater detail all those features that characterize the whole system rather than individual components. These may include the ease of assembly, aesthetics, ergonomics, safety, quality control, transportation, operation and maintenance, all of which have a bearing on the quality and cost of the product.

Generally, an assembly drawing is produced to enable the assessment of the product features enumerated above and to facilitate the detailed design of the various components and the selection of bought-out parts from supplier's catalogues.

(8) **Detail Design**

Having synthesized a satisfactory assembly for the product, the objective at the detail design phase is to prepare final instructions about the layout of the whole system, and the form, dimensions, surface properties and material selections for all the components. Apart from the features imposed by the assembly design, a particularly important consideration at this stage is the availability of the necessary manufacturing facilities and materials.

Depending on the type of product and manufacture (i.e. one-off, batch or mass production), assembly and quality control instruction may have to be produced for the production department as well as operating and maintenance documentation for the eventual customers.

(9) Production Planning

Having completely defined the configuration of the product, the physical means for its manufacture must now be planned. In an established enterprise, production planning will mainly consist of process planning and operations planning. Process planning involves the determination of the 'routing' for each component as well as the assembly requirements. In operations planning, each workcentre and operation involved is examined in turn to establish e.g. tooling requirements, optimum layout of workcentre, and the possible division of each operation into a series of elements.

Production planning is influenced to a great extent by the 'make or buy' policy of the enterprise and the required production volume.

(10) **Pre-production**

At this stage, both the product configuration and the production plans have been specified but not yet fully tested. The objectives at this stage are therefore (a) to test the adequacy of the production planning and (b) provide production samples for testing.

To obtain realistic test results, it is essential that pre-production is carried out according to a schedule and under conditions similar to those likely to prevail during full production.

(11) **Product Testing**

The economic viability of a product eventually depends on how acceptable it is to the intended market. The objective at this stage is therefore to check that the product performs according to the design specifications and to assess the quality of the product. In general, testing can be carried out at three consecutive levels as follows:-

- Bench testing where the 'loading' is artificially increased in order to show up the product weaknesses in the shortest period of time.
- Field trials where the product is released to a sample of the intended market. This is carried out mainly to assess quality of the product. (Quality is based upon the customer's actual experience with the product, measured against his requirements).
- Life trials where the product is tested on a life basis. These tests provide details of service requirements to ensure that a satisfactory performance can be maintained over a sufficient time period.

(12) **Full Production**

The testing and pre-production stages will normally reveal the necessity to make changes both to the design of the product and the production planning. Once these changes have been effected, full production can then commence. The objective at this stage is, through 'Production Control', to ensure the continued production of defined quantities of the product against a time scale and in accordance with a marketing strategy.

Although the relative importance of the modules will vary in different situations, all the modules shown are considered relevant regardless of type of discrete product, the size of firm, the volume of production or the number of firms involved in the process.

Fig 3.1 shows a definite start and end to the product realisation process. However, the dynamic nature of modern society means that its requirements with regard to the performance of existing products keep changing at an ever increasing rate. Consequently, product development work is a continuous activity in modern industry. In a sense then, the product realisation process can be viewed as a closed loop, with the market providing the link between the end and the start of the process.

3.6 Information transformation during the product realization process

The concept of information as a valuable resource and the general role of information in the everyday functioning of a manufacturing enterprise have been discussed in Chapter One. However, for the study of information processing in an integrated CAE system, it is necessary to model the flow of information directly related to product realisation, for a manufacturing enterprise. The general methodology outlined in Sect 3.6 provides a suitable framework for such a model. The development of an information transformation model is outlined in this section.

3.6.1 Classification of information relevant to the process of developing a product

To support the process, an enterprise needs to maintain and communicate many different types of information. **Lillihagen [23]** divides the data resource of a manufacturing enterprise into three main categories as follows:-

(a) **Engineering data**

This is data about the resources of the enterprise e.g. materials, manpower, machinery, industrial standards and know-how etc., that are valid beyond a specific product.

(b) **Product modelling data**

Product modelling data is that which enables the at-all-times correct description of the product under development. Examples are data relating to the geometry of components, the design specifications and logic, production planning and marketing forecasts. Once the product realisation process is completed, a certain fraction of the product modelling data becomes redundant and only data that is required during full production or for product development work need be maintained on a more permanent basis.

(c) **Information adminstration data**

The growth in the multiplicity and volume of enterprise data with time makes it imperative to create and maintain records to enable efficient adminstration of the various media e.g. specification documents, standards, drawings etc., on which information is stored. Such data about the data resource itself is here referred to as information adminstration data.

The types and quantities of data kept will, of course, vary from enterprise to enterprise, depending on factors such as the size and type of organisation and the product spectra.

3.6.2 Information flow at a module of the product realisation process

For product evolution to progress smoothly, there has to be efficient communication of information among the various activity modules. Taking the discrete-product realisation process to be a sequential one as discussed in Sect 3.5.2, the flow of information at a typical module can be modelled as depicted in Fig 3.2.

In Fig 3.2,

IN_1 is that product modelling data from the previous module on which the current module is directly

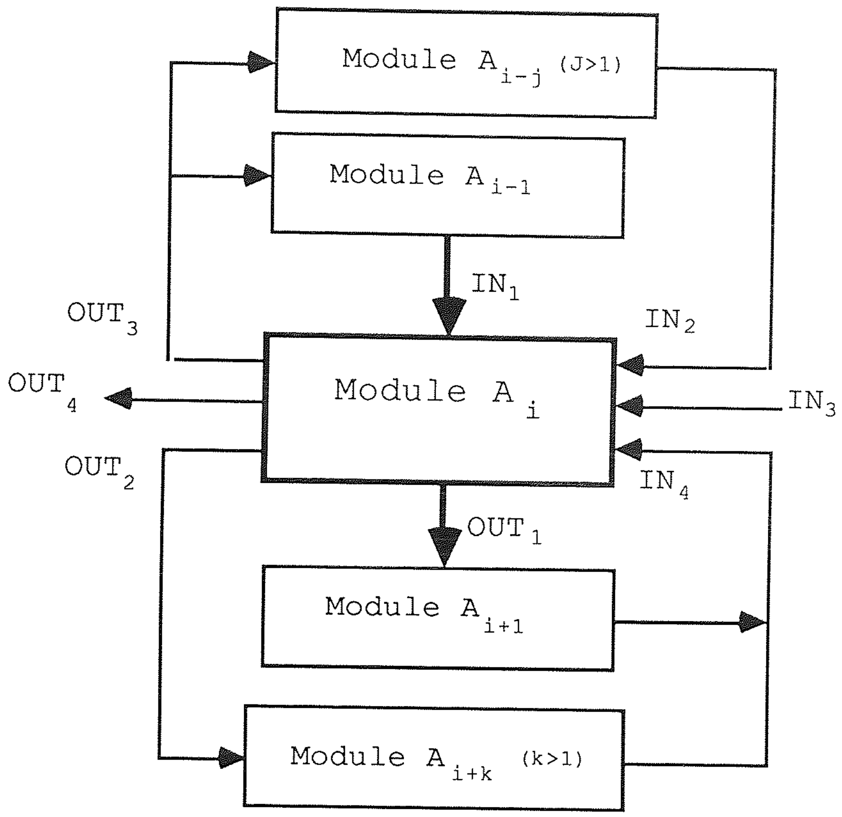


Fig 3.2 Depicting the flow of information at a module of the product-realization process

based.

IN_2 is product modelling and information adminstration data generated by earlier modules, that is relevant to the current module.

IN_3 is engineering data that is relevant to the current module.

IN_4 is feedback product modelling and information adminstration data from later modules in the case of iteration.

OUT_1 is that product modelling data generated by the current module, on which the next activity is to be directly based.

OUT_2 is product modelling and information adminstration data generated by the current module that is relevant to some later modules.

OUT_3 is product modelling and information adminstration data generated by the current module that is fed back to earlier modules in the case of iteration.

OUT_4 is product-modelling data that is judged to be applicable to other future products and is therefore adopted as an internal standard.

In the proposed methodology, the output A_{i+1} to the next module is evaluated for completeness and quality. Then, either the next activity is started or the process reverts to an appropriate earlier module, depending on the outcome of the evaluation.

3.6.3 **Analysis of information transformation requirements**

The information flow model shown in Fig 3.2 was used to analyze the information transformation requirements at each of the twelve activity modules in the proposed methodology. The information transformation requirements were synthesized from literature relevant to each module.

It was found convenient to adopt the use of Input-Process-Storage-Output (IPSO) charts in the presentation of the information transformation model. An IPSO chart is therefore shown for each of the twelve modules shown in Fig 3.1 (Fig 3.3.to Fig 3.14). These are placed at the end of this chapter.

In the IPSO charts shown, the 'Input' and 'Output' statements given are considered to be general and fairly rigid. On the other hand, the descriptions in the 'Process' and 'Storage' blocks are meant to be flexible and dependent on the type of product and enterprise involved. In formulating the contents of these two blocks, the intention was to indicate rather than enumerate the processing and information storage requirements for an activity. Any enumeration shown is mainly for clarity of description.

3.7 **Conclusion: Information as an enterprise resource**

A systematic approach to discrete-product realisation has been proposed in this chapter. The methodology proposed seems to meet satisfactorily all the general specifications put up in Sect 3.5.1 and its adoption can therefore be expected to yield the benefits listed in Sect 3.3.

For the purposes of the study of information processing in integrated CAE systems, the work presented in this chapter provides two important benefits as follows:-

(a) The need for an efficient information system to support product evolution is firmly established. An important inference from the work is that information is a valuable resource of an enterprise comparable to others like money and materials with the same characteristics of value, cost and scarcity and therefore needs to be managed efficiently. This notion has become increasingly significant as the technical ability to exploit data has expanded through the use of computers and associated communication systems.

(b) By putting a structure to the flow of information in a manufacturing enterprise, the proposal has helped to broaden the range of issues that should be taken into consideration when building up an information processing system. These include the quality of information (e.g.reliability, volume, form etc.,), user requirements at different phases and the complexity of the product under development.

The ready availability of a wide range of comprehensive and problem-oriented information at different phases of the product realization process is of the utmost importance. The methodology proposed in this chapter therefore provides a proper foundation on which to base the study of the information processing function in a CAE environment.

INPUT

Recognition by the Management of an enterprise of a need to initiate a new product. This can be stimulated by various factors (see Sect 3.5.4 (a))

PROCESS

1. Analyse the current state of the enterprise, particularly with regard to turnover and profits and set the objective to be achieved by the development of a new product
2. Identify market needs from sources such as:-
   (Internal) - Sales force, R & D, present customers etc.,
   (External) - Commercial labs, advert. agencies, patents office, firms going out of business etc.,
3. Guided by the corporate objectives, select a product area that the enterprise is is judged capable of exploiting fruitfully.

STORAGE

1. List of many other market requirements and product ideas for future use.
2. Data about target markets and potential sales volumes

Information concerning the capabilities of the firm as characterized by, for example, manpower skills & experience, availability of investmant capital, machinery & equipment, raw materials etc.,

OUTPUT

A general statement describing (a) the company objectives and product field and (b) the main functional requirements for the product that would satisfy the needs of the intended market

Fig 3.3 Showing the IPSO chart for the recognition of a need for a product

INPUT

A description of what the basic function of the product should be in order to fulfill the requirements of the intended market

PROCESS

Carry out a market survey to discover what features the potential customers want the product to embody and are willing to pay for.

Procedure:-

1. Plan the market research i.e determine the kinds of facts that are required to enable a proper assessment of the market.
2. Collect the required information by using direct approach methods, (interview, questionnaires etc.,), observational inquiry and use of published market data.
3. From the facts collected, determine the main product features required by the market.

STORAGE

1. Information gathered about the target market.

1. A checklist of major product features required by the market such as price range, bulk size etc.,
2. Approximate size of potential market.
3. Potential competing products.

1. Information from internal records about past market trends for 'similar' products.

OUTPUT

A broad description of the 'type' of product (solution) that the market will accept as adequately fulfilling its requirements.

Fig 3.4 Showing the IPSO chart for product selection

INPUT

A broad description of the type of product that has been assessed to be what the market requires.

STORAGE

1. Market requirements, mostly in qualitative form.

1. A record of the completed design specifications.

1. Relevant industrial guide-lines and standards.
2. Information concerning the current stage of the techno-logy in the relevant field(s).

PROCESS

Knowledge and experience are applied to the broad description of the product, to prescribe the required product features more quantitatively. These featues represent the minimum that should be achieved in design and manufacture if the requirements of the market are to be met.

Procedure:-

1. Specify all relevant 'inputs' to the system.
2. Specify all required 'outputs from the system.
3. Specify relevant features concerning the working environment.
4. Attach a 'measure of value' to each specification.

OUTPUT

A complete set of performance requirements for the product, stated in quantitative form as much as possible.

Fig 3.5 Showing the IPSO chart for the formulation of design specifications

INPUT

Design specifications i.e. a detailed description of the required features and performance of the end product.

PROCESS

In many cases, the same set of specifications can be satisfied (not equally) by products designed using fundamentally different concepts. The purpose at this stage is therefore to determine the best concept from a number of alternatives.

Procedure:-

1. Determine a set of candidate concepts for the design. Both new and existing concepts are considered.
2. Compare these alternatives use the specifications as the criteria.
3. Use a combination of formal selection techniques, intuition and experience to select the 'best' concept for the job at hand.
4. Carry out a 'concept check' to make sure that the concept chosen can infact be adopted for the design, taking into account all conceivable constraints.

STORAGE

1. Records of all concepts examined for possible use in future jobs.
2. List of other possible constraints not included in the specifications but which may affect the feasibility of the product.

1. Records of established concepts in the field.
2. Knowledge of techniques that can aid the synthesis of new concepts for the design i.e. heuristics.

OUTPUT

A description of the main features of the chosen concept.

Fig 3.6 Showing the IPSO chart for conceptual design

INPUT

A description of the main features of the concept chosen for the design.

STORAGE

1. Possible suppliers of sub-systems and components.
2. Detailed listings of the production facilities that may be required, including manpower and materials.

1. Company standards, including the make or buy policy.
2. Materials data.
3. Information on subsystems and components from alternative suppliers.

PROCESS

In many cases, the same concept chosen can be translated into more than one tangible form. The purpose at this stage is therefore to determine the optimum shape, size and materials (s) for each component or subsystem.

Procedure:-

1. Identify the major components and subsystems of the product as implied by the design concept adopted.
2. Generate alternative configurations for each of the subsystems and components. For buyable items, consult a number of supplier's catalogues.
3. Use a combination of formal analysis techniques, intuition and experience to determine the optimum combination of the various components and subsystems.
4. Carry out an economic feasibility study.

OUTPUT

A broad descrition of the configuration for the product, having considered its production feasibility and commercial viability.

Fig 3.7 Showing the IPSO chart for the production of system schemes and specifications.

INPUT	PROCESS
INPUT Descriptions of the proposed forms for each of the components and subsystems.	**PROCESS** At this point, the major system components have been given a form and the layout of the whole system has also been established. A thorough design analysis can therefore be done based on the more accurate loading, materials and environmental data now available. Procedure:- 1. For bought-out parts, make exact selections using the more detailed information now at hand, paying due attention to any user instructions given by the suppliers. 2. For parts being designed within the enterprise, carry out the appropriate calculations and make any modifications to dimensions and choices of materials as necessary in order to ensure the required performance.
STORAGE 1. Design specifications. ⇨ 2. Technical information concerning bought-out parts. 1. Maximum loading and environmental conditions to avoid physical failure of the various parts and subsystems. ⇦ 1. General materials data. ⇨ 2. Company standards e.g. safety factors, analysis procedures etc.,	**OUTPUT** Design analysis documentation and revised schemes and specifications.

Fig 3.8 Showing the IPSO chart for design analysis

INPUT

Complete system schemes and specifications and full details about all bought-out parts.

STORAGE

1. Details of any space limitations.
2. Design specifications on non-functional features.
3. Information on other desirable features which may not have been included in the formal specifications.

1. Exact technical specs. for all bought-out parts.
2. Exact dimensional and surface texture limits for parts that are to be manufactured within the company.

1. Standards for equipment in the particular field e.g. as regards safety, noise levels etc.,
2. Ergonomic data and standards.

PROCESS

The functional design of the system is considered to be completed at this stage. The purpose is now to transform the scheme into a feasible assembly.

Procedure:-

1. Check all 'mating' features on bought-out parts for compatibility.
2. Use the geometry of the bought-out parts and any space limitations to specify dimensional limits for the parts to be manufactured within the enterprise.
3. Consider relevant non-functional aspects of the assembly e.g. aesthetics, ease of manufacture, maintenance, transportation etc.,
4. Prepare an assembly drawing and a parts list.

OUTPUT

1. Assembly drawing.
2. Complete specifications for all bought-out parts.

Fig 3.9 Showing the IPSO chart for assembly design

INPUT

Assembly drawing, complete specs. for all bought-out parts, dimensional and surface texture limits for all parts to be made within the firm.

STORAGE

1. Relevant design specs. e.g. weight limits, required maintenance frequencies etc.,

1. Stock material requirements
2. Manpower, machinery, tooling, handling and other production requirements.

1. Existing manufacturing capability information as characterized by manpower skills and numbers, machinery etc.,
2. Standard sizes of stock material, industrial standards for components if any, materials data etc.,
3. Drawing standards e.g. BS 308

PROCESS

The objective of detail design is to create engineering drawings and instructions which can enable individual parts of an engineering system to be manufactured.

Procedure:-

1. Identify all functional features of each component and attach appropriate tolerances.
2. While preserving all functional features, carry out a form design of each comp0-nent, taking into account all other important features such as method of production, inspection, handling, use of standard stock materials , standardization etc.,
3. Prepare production drawings as necessary.

OUTPUT

Production drawings for all parts that are to be made internally or 'contracted out'.

Fig 3.10 Showing the IPSO chart for detail design

INPUT

Assembly drawing, detail drawings and a complete list of bought-out parts and subsystems.

STORAGE

1. Corporate policy and objectives.
2. Quality control standards.
3. Details of existing resources such as machinery and man-power.

1. Details of all resources required in order to achieve the objectives.
2. Details of information processing requirements.

1. Details of appropriate production and inspection technology.
2. Relevant legal considerations e.g. Factories Act, Trade Union law etc.,

PROCESS

The objective of production planning is to plan the physical means to be used by the firm to produce an item.

Procedure:-

1. For each component, choose the material form with which its production is to start.
2. For each component, determine the series of work tasks required to change its state into the finished form i.e. 'routing'.
3. Determine the production facilities required to convert the stock material into finished components.
4. Make choices of handling and storage systems for items around the enterprise.
5. Decide on the layout of the various machinery and equipment in the available area for production.

OUTPUT

Complete production plans which, if implemented would enable production of the items, consistent with the firm's objectives and policy.

Fig 3.11 Showing the IPSO chart for production planning

INPUT

Complete production plans.

STORAGE

1. Pre-production targets, in terms of volume and time.
2. Quality assurance standards.
3. Pre-production schedule.

1. Results of a 'time and motion' study
2. List of manufacturing and procurement problems encountered during the pre-production run.

1. Technological information

PROCESS

Pre-production is carried out for two main reasons i.e. (a) to manufacture sample items for testing and (b) to test the soundness of the production plans.

Procedure:-

1. Purchase and install the essential manufacturing plant including testing equipment.
2. Design and manufacture or purchase essential tooling.
3. Order raw materials and all bought-out parts
4. If necessary, initiate training programmes for the production personnel.
5. Manufacture pre-production batch according to a schedule

OUTPUT

Pre-production samples for testing.

Fig 3.12 Showing the IPSO chart for pre-production

INPUT

A batch of products for testing

PROCESS

The objective at this stage is to test the technical performance of the product as well as its acceptability to the intended market.

Procedure:-

1. Bench-test part of the pre-production batch.
2. If the results of the bench test are satisfactory, then conduct field trials using a representative cross-section of the target market.
3. Commence life trials to obtain details of the longer term service requirements as well as ideas for product improvement.
4. Use the experience gained during pre-production and testing work to update both the design of the product and the production plans as necessary.

STORAGE

1. Test specifications. ⇨

1. Complete data to allow the drawing up of 'technical specifications' for the product. ⇦
2. Guidelines for future product development.

1. Testing standards i.e. standardized procedures and figures if any, appropriate for the type of product. ⇨

OUTPUT

Confirmation of both the 'quality' of the product and the adequacy of the production plans to allow full production as required.

Fig 3.13 Showing the IPSO chart for testing

INPUT

Confirmation of both the quality of the product as well as the adequacy of the production plans to sustain full production.

STORAGE

1. Market information.
2. Precise information on the capabilities of the various production resources.

1. Production control information.
2. Ideas for product improvement and new product development.

1. Formal production scheduling techniques.
2. Formal production control techniques.

PROCESS

The aim at this final stage is to initiate and maintain the production of defined quantities of the product against a time scale and in accordance with a marketing strategy.

Procedure:-

1. Implement all necessary changes to the production plans, including those partaining to manpower.
2. Carry out production scheduling.
3. Establish packing and despatch procedures.
4. Motivate the sales force and and service network.
5. Commence full production.
6. Formally launch the product.

OUTPUT

A product is released to the market, intended to satisfy the requirements identified at the beginning of the realization process.

Fig 3.14 Showing the IPSO chart for full production

C H A P T E R   F O U R

INTEGRATED COMPUTER AIDED ENGINEERING SYSTEMS

4.1 Introduction

Computers are now in common use in both scientific and commercial fields. New electronic components such as integrated circuits have led to the development of computers that have considerable computing power and which are physically small, reliable and sufficiently low in cost to make their use acceptable in a larger number of application areas.

The product realization process outlined in Chapter 2 can be carried out without the aid of computers. However, computers offer three main facilities which make their utilization desirable [**24**]. These are (a) their ability to perform complex or tedious calculations quickly and accurately, (b) their ability to manipulate and store large amounts of data and (c) their ability to present information graphically. A judicious application of computer technology can therefore improve the productivity of the process by, for example, enabling the reduction of product development time and the improvement of product quality.

A brief discussion of Computer Aided Engineering (CAE) and, more importantly, 'integrated' CAE is presented in this chapter. Working definitions of both CAE and integrated CAE are given, each followed by a brief state-of-the-art review. After making the assertion that truly integrated CAE systems have yet to be achieved, a possible strategy for their achievement i.e. the adoption of the database concept to the management of engineering information, is proposed at the end of the chapter. The proposal is elaborated in the next two chapters.

4.2 Computer Aided Engineering

The increased use of computers in engineering has led to a proliferation of terms used to describe computer aided activities. Examples are Computer Aided Design (CAD), Computer Aided Manufacture (CAM) and Computer Aided Production Planning (CAPP). Unfortunately, there is also a lack of consensus on the exact meanings of these terms.

The definition of Computer Aided Engineering (CAE) adopted in this research work is that offered by **Kellock** [25]. He defines CAE as the application of computers to the production of components and finished products from design through to final inspection, assembly and despatch to the market. CAE is then taken to embrace not only CADCAM which, generally speaking, relates only to the generation of component geometric data and numerically controlled machining, but also the other essential activities in the product realization process such as design specification and production planning.

The term 'Computer Aided Engineering' does not designate a new field of engineering activity but only implies that more efficient methods and tools are being used in the practice of engineering. **Jurocic** and **Barr** [**26**] in fact contest the use of the term altogether. They contend that since it is normal that new tools and methods are incorporated into engineering as they become available thus creating contemporary modern engineering, the classification into computer aided or non-computer aided engineering should really not arise at all.

4.2.1 **Computer Aided Engineering versus Automation**

It is important to distinguish between CAE as defined above and the notion of the automation of the product realization process through the use of computers. While automation implies the replacement of the human operator with a set of computer-based tools, the primary idea in the CAE field is the augmentation of the productivity of the human operator by, for example, speeding up the performance of low level detailed work and, admittedly, providing automated generators for specific portions of the operator's work. By this token, most currently so called automated manufacturing systems are actually CAE systems and, until algorithmic procedures are established for all the stages of the product realization process (see Sect 2.5), complete automation of the process will remain only an ambitious research topic.

4.3 CAE - a brief state-of-the-art review

4.3.1 Computer Aided Engineering tools.

The product evolution process as outlined in Chapter 2 can be said to comprise a stepwise specification of tasks, each followed by a synthesis of candidate solutions, an evaluation of these and then a choice of an optimum solution, until the product eventually takes a concrete form. Accordingly, current CAE tools can be classified into three broad groups i.e. those for synthesis, those for analysis and those for information management.

(a) **Synthesis tools.**

The ultimate goal of any synthesis tool is to transform a higher level description of an object or system into a more concrete form, amenable to analysis and further development.

One large class of synthesis tools is that for design capture. For example, a drafting system enables a designer to communicate his design description to the computer after which other tools can then be used to analyze and manufacture the item. Other synthesis tools are those that aid the mapping of a higher level description of an object into its physical form (e.g. NC machining systems) and those for placing and routing (e.g. Production

Planning systems).

An expanding and interesting class of synthesis tools are those based on applied artificial intelligence research called 'Expert Systems'. These tools attempt partially to automate the reasoning of an expert in transforming a behavioural specification into several reasonable alternatives and then leave it to the user to make a choice from these.

(b) **Analysis tools**

Analysis tools are used to check the correctness of the proposed design. In the design of mechanical objects, the main analysis tools of interest are of three types as follows:-

- Those that assist the determination of the static and dynamic behaviour of components or subsystems. A common example is Finite Element software for determining the stresses in and deformation of a component under load.
- Simulator tools. These model the input and output behaviour of a system and therefore help to uncover communication problems that may arise through faulty interaction of the subsystems in real time.
- Topological analyzers. These help to check the correctness of the layout or assembly of components and subsystems in space.

(c) **Information Management tools**

These tools are responsible for creating and maintaining a correct description of the product at all times and therefore form the foundation upon which synthesis and analysis tools can be built. The most common element in every current information management tool is the storage of data in files on disc or tape.

The important role of information transformation in the product realization process has been discussed in Chapter 2. Therefore, of the three groups of CAE tools just described, the last one is of special interest to the research work presented in this thesis. It is perhaps the most important and certainly the least understood aspect of a CAE system.

4.3.2 Current computer application in the product realization process.

In general, the share of activities that can be expressed in algorithmic form and therefore enable their computerization increases as the product realization process advances as depicted in Fig 4.1.

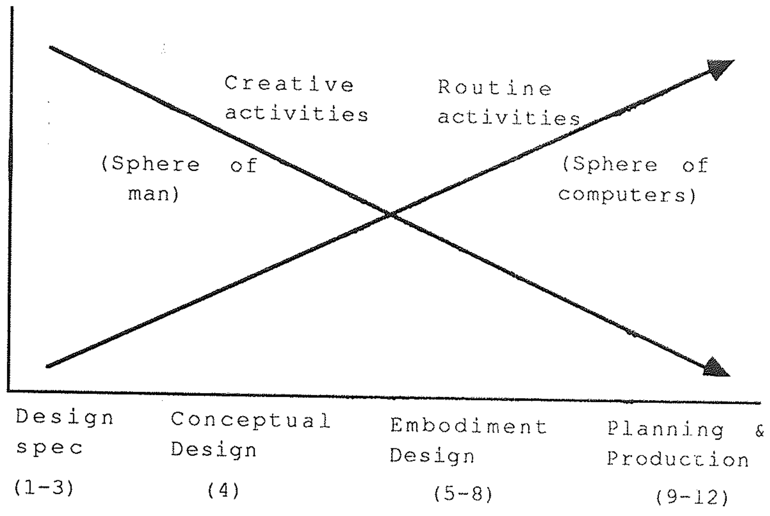


Fig 4.1 Depicting current application of computers in the product realization process

(The numbers shown in brackets refer to the corresponding stages in the general product-realization methodology proposed in Chapter 3)

The activities within the product specification and conceptual design phases require familiarity with the type of product, experience and creativity and can therefore generally not be expressed in algorithmic form. Currently, substantive computer application starts at the embodiment phase when various synthesis tools such as draughting packages and solid modellers and analysis tools such as Finite Element packages are used iteratively until a satisfactory description of the product is established. The beginning of the production phase requires some creativity as well but the availability of both the product modelling information and the necessary technological information at this stage means that more realistic computer aids can be developed for the subsequent phases.

Although the development of CAE tools for all the activities is both feasible and advantageous to create and maintain an information system that spans the whole process. Such a system would bring about the the following important benefits:-

(a) The non-computerized activities can still benefit from the availability of an information depository through a query facility.

(b) The expert systems approach promises to enable the application of computer technology to a greater number of activities. The performance of expert systems depends to a large extent on the availability of a wide range of information.

(c) The feasibility of integrating current and future CAE tools would be enhanced.

The issue of integration of CAE tools is discussed in greater detail in Sect 4.4 and Sect 4.5.

4.4 **Integrated Computer Aided Engineering Systems**

By and large, Computer Aided Engineering is still a fragmented technology in which computers are used to aid discrete activities such as draughting or Finite Element analysis, sometimes linking these as subsystems. Over the last few years however, the move towards integration of CAE tools has been boosted by the greater need to improve productivity and by major developments in the enabling technologies such as computing and storage, database management and communication network architectures.

The concept of an integrated CAE system, its potential benefits and the current problems hindering its implementation are briefly discussed in this section.

4.4.1 **Integrated CAE - a definition**

CAE as defined in Sect 4.2 can be implemented in an uncoordinated manner where several standalone computer systems are used to assist individual activities or departments within an enterprise. This is generally what leads to the so called 'islands of automation' in today's manufacturing organizations. An integrated CAE system is one in which the various tools are linked together thereby facilitating better management of the information resource in the firm. The goal of an integrated CAE system (also alternatively termed a Computer Integrated Manufacturing System (CIMS)) is the optimization of the total business over and above that of the individual components.
Integrated CAE is the underlying philosophy that is required in looking forward to the fully automated factory of the future. Selected elements of an integrated system might be CAD, CNC manufacturing, Computer Aided Production Planning (CAPP), Database Management Systems (DBMS) , Expert Systems, Flexible Manufacturing Systems (FMS), Material Requirements Planning Systems (MRPS), process control systems and Robotics.

Logically, there are two major components of an integrated CAE systems; hardware and software. The emphasis in the research work presented was in the integration of the software tools in a CAE system.

4.4.2 **Potential benefits of a truly integrated CAE system**

The importance of information flow in a manufacturing enterprise has been discussed in Chapter 2. Total system integration would enhance the efficiency of information flow by reducing the clerical effort in data capture as well as minimizing errors in calculations and data transposition.

Another important benefit envisaged is the enabling of faster responses to product design or manufacturing changes with better control of such changes. Others include the provision of more detailed management and control information and the reduction of computing costs [**28**]. The overall benefits from a truly integrated system would be reduced manufacturing costs, higher productivity, shorter lead times and improved product quality [**28,29**].

4.4.3 **Problems hindering successful CAE system integration**

Although the concept and potential benefits of integrated

CAE systems have been promoted and recognized respectively over the last few years, a worldwide survey conducted by Ingersoll Engineers [28] showed that integration of businesses through the widespread and dedicated use of computers hardly exists at all. Clearly, there still are significant barriers to their successful development and implementation.

The factors hindering successful integration can be shown to exist at two levels; industrial and enterprise.

(a) **Problems at industrial level**

There are two main problems at this level as follows:-

(1) There is a lack of consensus as to who should carry out the integration. Turnkey system vendors claim that users want entire systems integrated by the vendor because it can be difficult to get a number of products from different sources to work together and the expenses involved might be too great for some users to handle. On the other hand, vendors of standalone workstations and third party software argue that no one vendor can ever supply one hundred percent of the items that a particular user may require.

Since there are more possible applications than the turnkey vendor can ever cater for, a compromise solution would be the development of open-ended turnkey systems that can get the user productive quickly and still let him add more modules later by perhaps turning to the third party market place.

(2) There are inadequate industrial standards for software interfacing. This is partially due to vested commercial interests by some current major turnkey system vendors. Ingersoll Engineers [**28**] make the claim that vendors frequently and quite deliberately develop individual characteristics into their products in an attempt to lock customers into their range.

An example of the type of software standards required is the Initial Graphics Exchange Specification (IGES) [**30**], a data interchange specification developed by the U.S. armed forces and NASA. IGES is aimed at enabling geometric data to be readily exchanged between different computer systems.

(b) **Problems at enterprise level**

These can be classified into three main groups (1) those caused by improper perception of integrated CAE by managements (2) constraints imposed by existing practices and plant and (3) technical problems.

(1) **Improper perception of integrated CAE by management.**

This is identified by many researchers [**1,28,31**] as the major cause for the widespread dissatisfaction with current CAE-oriented products. The main problem seems to be the failure by managements to view integration as a corporate objective that requires a definite strategy for its achievement. This normally leads to a piecemeal implementation of CAE, normally grafted onto traditional and probably outdated manufacturing practices. Another problem is the mistaken view partially promoted by trade literature that integrated CAE is one powerful package of computer technology that can almost automatically rejuvenate an ailing business.

Other problems arise from unsubstantianted expections about complexity and cost and lack of education concerning the rational steps to take in the implementation of advanced manufacturing technology.

(2) **Contraints imposed by existing practices and plant.**

In introducing integrated CAE in a going concern, there are several issues and realities that management is faced with. **French** [**32**] provides a comprehensive list of the problems that can arise. The more critical ones are as follows:-

- Very few companies can afford the capital investment necessary to make existing plant compatible with an integrated system.

- The existing forms of functional organization, reporting procedures, control etc., may be completely incompatible with the requirements of an integrated system. A complete reorganization of the enterprise may therefore be needed.
- The competitiveness of modern industry generally requires that an enterprise keeps up with advances in the design and production processes for specific products. On the other hand, the decision to commit extensive resources to CAE-related products usually carry the assumption that the process itself is assured of a long life cycle.

Other problems include a shortage of technical skills in the CAE area and the hostility to integration that might exist at various levels in the organization.

(3) **Technical problems**

The main technical problems in introducing integration in a non-integrated environment arise from the lack of compatibility among the various CAE elements. This occurs on two levels:-

- Incompatibility between various stages of CAE or islands of automation. For example, the existence of a drafting suite and CNC software on the same computer system but which are unable to share data, perhaps because they are written in different high level languages.

- Incompatibility among systems from different manufacturers.

Many of the problems of incompatibility are caused by lack of planning for CAE system integration but they can also arise due to the high turnover of computer hardware and software in the market place.

4.5 Conclusion: Centralized management of information is the key to CAE system integration

The discussion in this chapter has shown that the integration of manufacturing systems through the use of computer-based technology is still in its infancy. At the same time, despite the problems outlined, it has become apparent in the last few years that the integration of CAE systems is clearly the direction required by today's manufacturing environments. The continued availability of high computing capacity at modest cost will enable integration to become an attainable objective.

As the use of computer based tools continues even in an unintegrated form, most of the problems outlined in Sect 4.4.3 are bound to become less and less significant and the main factor preventing integration will then be poor management of the information resource in the enterprise. A general recommendation from the researchers in the CAE area [28,33] is that integrated manufacture is approached from first principles by examining all aspects of a business,and not simply as a combination of existing technologies that must be linked together. Implicit in this approach is the need for a centralized management of information since information is the common denominator to all activities in an enterprise. However, it has to be acknowledged that this ideal is more feasible in new enterprises where the information flow can be planned with relative ease than in an existing ones where the necessary bottom-up analysis of the data may prove extremely difficult to carry out effectively.

The rest of the thesis is devoted to pursuing integration through the centralized management of the data resource in an engineering organization. More specifically, the 'database' concept, so far successfully applied to the management of commercial information is examined at length, with a view to establishing its suitability to the handling of information in a CAE environment. The domain for integration in engineering is very broad and cannot possibly be explored fully in any one research effort. The research on which this thesis is based was therefore focussed on the problems of integration in the design area while keeping the total environment in mind. The research showed the feasibility of using conventional database practices and technology (such as are embodied in proprietory database managenent systems), to achieve data shareability among the several application programs constituting a mechanical engineering design system.

CHAPTER FIVE

THE DATABASE APPROACH TO INFORMATION MANAGEMENT

5.1 Introduction

The discussions in Chapters 2 and 3 have established that a key requirement in any integrated manufacturing system is the efficient communication of information within and across the three main areas of activity namely (a) design and analysis, (b) production planning and manufacturing and (c) administration and business operations. One approach to system integration is therefore to adopt a system-wide management of the information resource whereby appropriate data is modelled centrally and stored in such a way that it is readily available to a multiplicity of applications. This is hereafter referred to as the 'database' approach.

The database concept and the current practices and technology associated with it in the commercial field are reviewed at length in this chapter in order to justify its consideration for application in engineering systems. This research work showed that provided certain special characteristics of engineering data are taken into account, these concepts and technologies can be applied successfully to the problem of integration in computer aided engineering systems. The issues involved are discussed at length in Chapter 6.

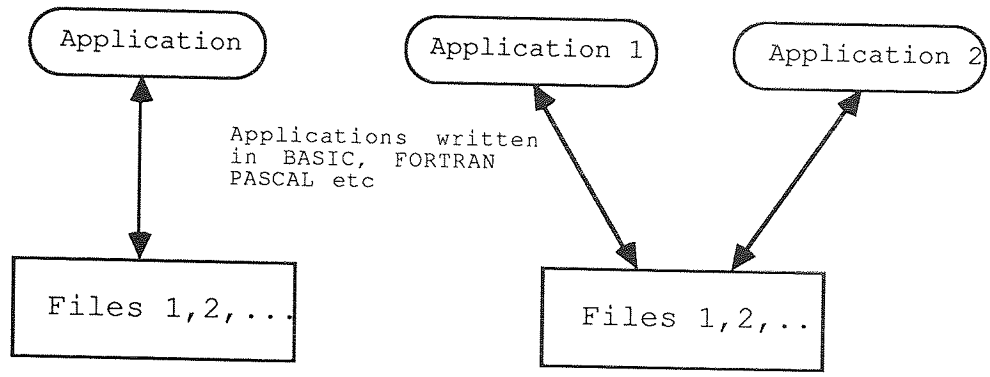
5.2 The database approach to information management

There are currently two contrasting approaches to information system design as depicted in Fig 5.1.

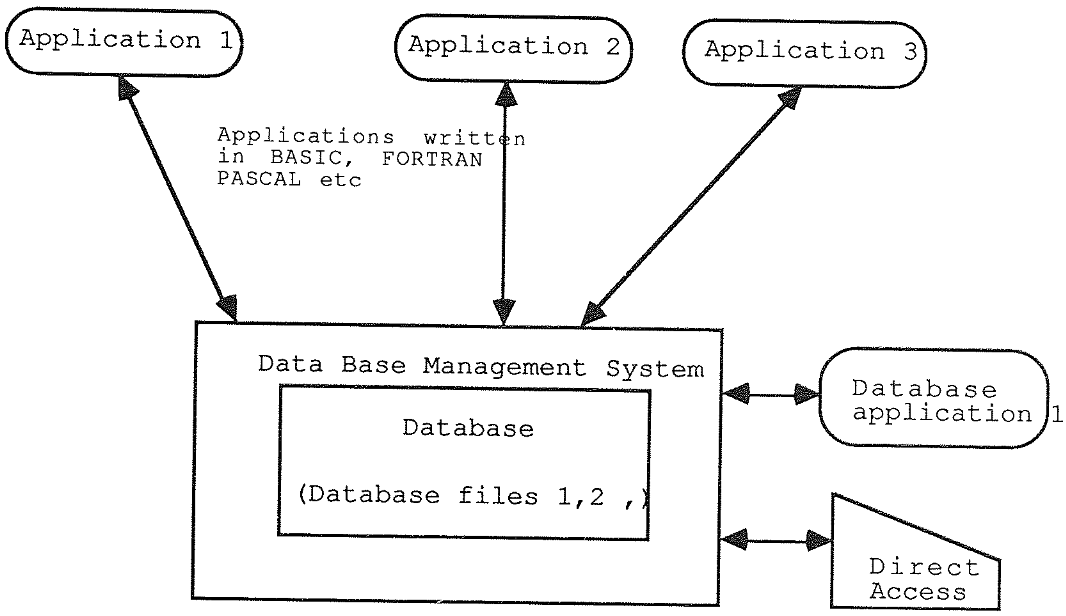
The first approach (Fig 5.1 (a)) focusses on the data processing needs of individual departments or application programs. In computer aided systems based on this approach, the file and record descriptions are embedded within each individual program. As a result, each program or set of programs 'owns' its data files and the logic of each program is very closely interwoven with the data formats and descriptions. This approach is still prevalent in CAE systems where data is still organized to support either an individual program or an 'island of automation'.

The database approach represents a different concept in information resource management. In this approach (Fig 5.1 (b)), data is viewed as an important shared resource that must be managed like any other asset such as manpower and machinery. This concept is rooted in an attitude of sharing common data resources and implies that the control of those resources is released to a common responsible authority.

The database itself is at the root of this approach. The most concise and widely quoted definition is that given by **Martin** [**64**]. He defines a database as a 'shared collection of interrelated data, designed to meet the needs of multiple



(a) Traditional file processing approach



(b) Database approach

Fig 5.1 Showing two contrasting approaches to information resource management

types of end-users. The items of data are stored so that they are independent of the programs that use them. A common and controlled approach is used in adding new data and modifying and retrieving existing data. A database is not only shared by multiple users but is also perceived differently by different users.'. A database is therefore more than a base for data; it is the basis of a new approach to data processing. Thus, the alternative spelling 'data base', in contrast to 'database', is misleading and is not used in this thesis.

The database concept first become popular in the mid-60's with the development of information management systems (IMS). Since then, database technology has developed steadily and has been successfully applied to data management in commercial environments. With the explosion in computer applications in engineering over the last decade, there is now an expanding interest in the possibility of applying the concept to engineering systems. However, the suitability of current formal database concepts and technology for application in engineering systems has yet to be established and currently constitutes an active research topic. The arguments about suitability are presented in Chapter 6: an important objective of the research work presented in this thesis was to contribute to the discussion. This contribution is presented in Chapters 8 and 9.

5.2.1 **Benefits of the database approach**

The database approach, as applied in commercial systems, is said to offer a number of important advantages compared to the traditional approach. The more significant are discussed below.

(a) Reduction of data redundancy.
Previously separate (and redundant) data files are integrated into a single logical structure. In addition, each data item occurrence is ideally recorded in only one place in the database. In this way, data redundancy and the wastage of storage space and data inconsistency that can be associated with it are minimised in the system.

(b) Data integration.
In a database system, data is organized into a single logical structure, with logical relationships defined between associated data entities. In this way, the user can easily relate one item of data to another.

(c) Data sharing.
A database is intended to be shared by all authorized users or application programs in an organization. Each functional department or application program is provided with its own view of the database which is a subset of the conceptual database model. These user views simplify the sharing of data since they provide each user with the particular view of the data required without making the user aware of the overall complexity

of the database.

(d) Enforcement of data standards, security and integrity.
Establishing the data administration function is an important part of the database approach. The data administrator group has complete jurisdiction over the database and is responsible for establishing data standards, and controls for accessing, updating and protecting data. If proper controls are applied, the data integrity and protection is improved compared to a dispersed data file system.

(e) Ease of applications development.
Once a database system has been established, the cost and time for developing new applications using the data is greatly reduced. This is because the application programmer is no longer preoccupied with designing and maintaining master files.

(f) Data independence.
The separation of the data descriptions from the applications that use them is called data independence. Within limits therefore, either the data or the applications programs that use the data can be changed without necessitating a change in the other factor.

This improves the maintainability of the whole information system and also reduces program maintenance.

Data independence is one of the major objectives of the database approach and is therefore elaborated in Sect 5.3.

5.2.2 **Possible demerits of the database approach**

Despite the benefits listed above the database approach has some disadvantages which may be significant in certain situations. Some of these are as follows:-

(a) The benefits listed above are, to a certain degree due to the coordination and standardization that is inherent in the database approach. Because of the larger user base for the same data, this coordination and standardization may result in greater difficulty and cost in fulfilling the requirements for confidentiality, data quality, and security.

(b) Increased computer costs due to the fact that practical database systems use standardized software (Database Management Systems) that is likely to be a little less machine efficient than the tailor-made programs of conventional systems.

(c) There may be other disadvantages such as longer response times for application programs due to the fact that the individual applications interfaced to the database are forced to adhere to certain standards and share certain resources.

The possible benefits and disadvantages of the database approach listed are either deduced from the concept or based on actual experience with database systems in the commercial field. Although their relative significance cannot be the same in all environments, they constitute a proper set of criteria that can be used to assess the success or failure of the application of the database concept in a particular environment.

5.3 Current database concepts, practices and technology

At a purely conceptual level, the application of the database approach to the management of engineering information seems both feasible and desirable. The arguments on suitability are however not at the conceptual level but relate to whether the existing formal database concepts (e.g. database models, design methodologies etc.,) and technology (e.g. Database Management Systems) can be successfully deployed in engineering systems. A brief outline of the structure of a database system is therefore presented in this section to precede the discussion given in Chapter 6 regarding the extension of conventional database concepts to engineering systems.

5.3.1 **The architecture of a database system**

Fig 5.2 shows the general architecture of a modern database system. The structure shown has evolved mainly from the proposals of two important bodies concerned with establishing standards in the data processing field. These are (a) the CODASYL DDLC (the Data Description Language Committee of the Conference of Data Systems Languages) [34], and (b) the ANSI/SPARC (the Standards Planning and Requirements Committee of the American National Standards Institution) [35] groups. The terminology used in Fig 5.2 and in the rest of the chapter is that of the ANSI/SPARC group.

There are two important concepts prevalent in conventional database practice i.e. (1) three views of data and (2) data independence. These are introduced below.

(1) **Three views of data**

Fig 5.2 depicts the three views of data necessary in a database system.

(a) **The external view**

The external view is the view of the data as

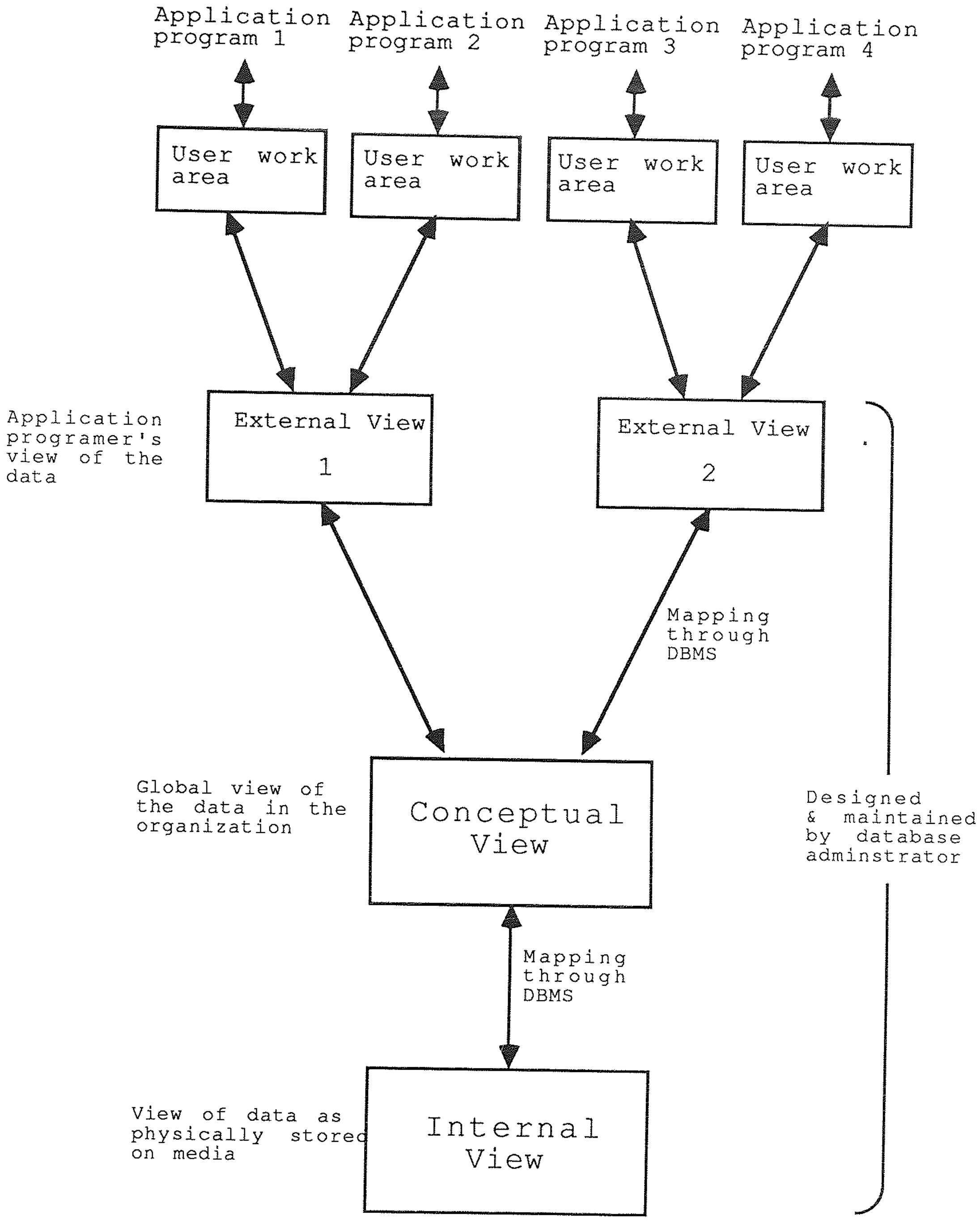


Fig 5.2 Showing the architecture of a database system

perceived by individual application programmers. It is a subset of the conceptual model and it is designed and formatted to suit an individual application. Any number of external views can can exist in the system and a number of applications can share the same external view. An implementation of an external view is termed a 'subschema'.

(b) **The conceptual view**

The conceptual view is the definition of the entire information resource at an abstract level. this includes all entities represented in the database, the relationships among its entities and other features relating to, for example, data integrity maintenance. The conceptual view does not contain any implementation details such as file organization, access methods or software and hardware requirements. An implementation of a conceptual view is termed a 'schema'. The schema is at the very heart of a database approach to information management and its optimal design and implementation is the most critical work module in the course of establishing a database system. Database design and implementation is discussed in Sect 5.4

(c) **The internal view**

The internal view is the view of the data as it is stored on a medium, including the method of access. By and large, this is determined by the combined characteristics of the Database Management System (DBMS)(see Sect 5.3.2) used in the implementation of the database sysm and is therefore generally outside the concern of a user of a proprietary DBMS.

Fig 5.2 shows that in a modern database system, the mapping between subsequent views is performed by the DBMS. This aspect is discussed in greater detail in Sect 5.3.2.

The concept of separate schema and subschema allows the separation of the description of the entire database from the description of the portion of the database known to an individual application. This concept yields two significant advantages as follows:-

- An individual programmer need not be concerned with the universe of the entire database, but only with those portions which are relevant to the application. This eases the writing, debugging and maintenance of application programs.
- A common language can be defined to create and manipulate the schema while allowing the subschema to be described in a manner oriented towards the language used in the application program using it. This permits the use of several languages e.g. BASIC, FORTRAN etc., chosen on the basis of their suitability to the application at hand, to process the same database.

(2) **Data independence**

The notion of data/program independence is central to the database concept. Ideally, two levels of independence are needed as follows:-

- Logical independence

  This refers to the ability to change the overall logical structure (the schema) without necessarily having to alter the application programs and vice-versa. The notion does not include the capability of the database system to automatically cope with changes in the algorithm of the program or its view of the data.

- Physical independence

  This refers to the ability to change the physical storage of the database or the access mechanisms without needing to change either the schema or the application programs

5.3.2 **Components of a database system**

The major components of a typical database environment and the interfaces among them are shown in Fig. 5.3. These are briefly described below.

(1) Application programs

These are programs coded in a variety of languages such as BASIC, FORTRAN etc., that use data from the

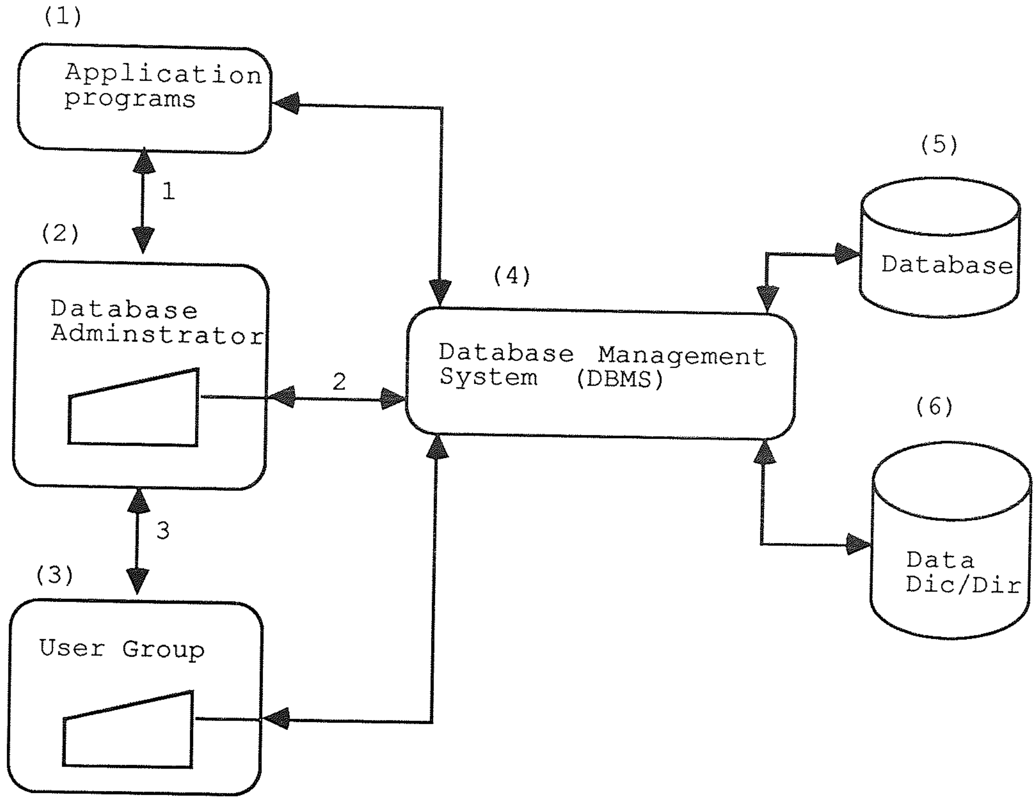


Fig 5.3 Showing the main operational entities and interfaces in a database environment

database. For example, the experimental CAE system developed in the course of this research work (described in Chapter 7) includes a rolling bearing selection program that uses bearings data separately stored on the database.

(2) The Database Adminstrator (DBA)

In the database environment, there is normally a need to assign the responsibility for the efficient management of the data resource to an individual or group in the organization. Typically, the duties of the DBA may will include the following:-

- the design, implementation and maintenance of the database system (interface '2' in Fig 5.3)

- assisting application system analysts and programmers with regard to data modelling and the interfacing of applications to the database (interface '1' in Fig 5.3) and
- assisting the user group with the formulation of queries to the database (interface '3' in Fig 5.3)

(3) User group

The user group consists of all requesters of data. With the increasing use of non-procedural data manipulation languages (DML), it is possible for end users to access the database directly i.e. without having to write application programs. An example is the use of the DML commands that are part of the 'DBASE II' DBMS.

There are three main categories of direct user transactions with the database: read only, add/delete and modify. Fig 5.3 shows that these are all performed through the DBMS.

(4) The Database Management System (DBMS)

Fig 5.3 shows that all access to the database is through the DBMS. The DBMS is a generalized software system which manages the database, providing facilities for organization, access and control. The term 'generalized' means that the DBMS is independent of individual applications. Examples of DBMSs presently available on the market as standard packages are DBASE II (used in the research work), RAPPORT, ADABAS etc.,

DBMSs typify the current state of database technology and are therefore discussed in greater detail in Sect 5.3.2.1.

(5) The database

This is the physical repository of all user data and corresponds to the internal view shown in Fig 5.2.

(6) Data Directory/Dictionary (DD/D)

The Data Directory/Dictionary is a repository of the metadata i.e. of 'data about data' stored on the database. As the name implies, a DD/D system has two logical sections. Briefly, the data dictionary contains definitions of records, data items, relations and other data objects that are of interest to users or are required by the DML commands of the DBMS. The data directory section, on the other hand, contains data about where data is on the database.

DD/D systems are key tools for managing the data resource. They are therefore discussed in more detail in Appendix 4 in conjunction with a description of the design and implementation of a DD/D system for the experimental CAE database system developed in the course of this research work.

In summary, the database environment depicted in Fig 5.3 is an integrated system of hardware, software and people that is designed to facilitate the creation, storage, retrieval and control of the information resource.

5.3.2.1 **Database Management Systems**

A database managenent system (DBMS) is the data processing system providing the means to access, organize and control all information stored in the database. Three important issues regarding DBMSs are discussed in the subsections (1) to (3) that follow.

(1) **DBMS software components**

A typical DBMS has two operational modules: a 'database control system' and a 'database storage system'. The control system accepts calls for data from user programs and examines the conceptual and external schemas to determine what conceptual records are required to satisfy the request. A call is then placed on to the storage system which manipulates the underlying storage files to satisfy the request. This is shown in Fig 5.4.

(2) **DBMS functions**

The essential feature in a database system is the

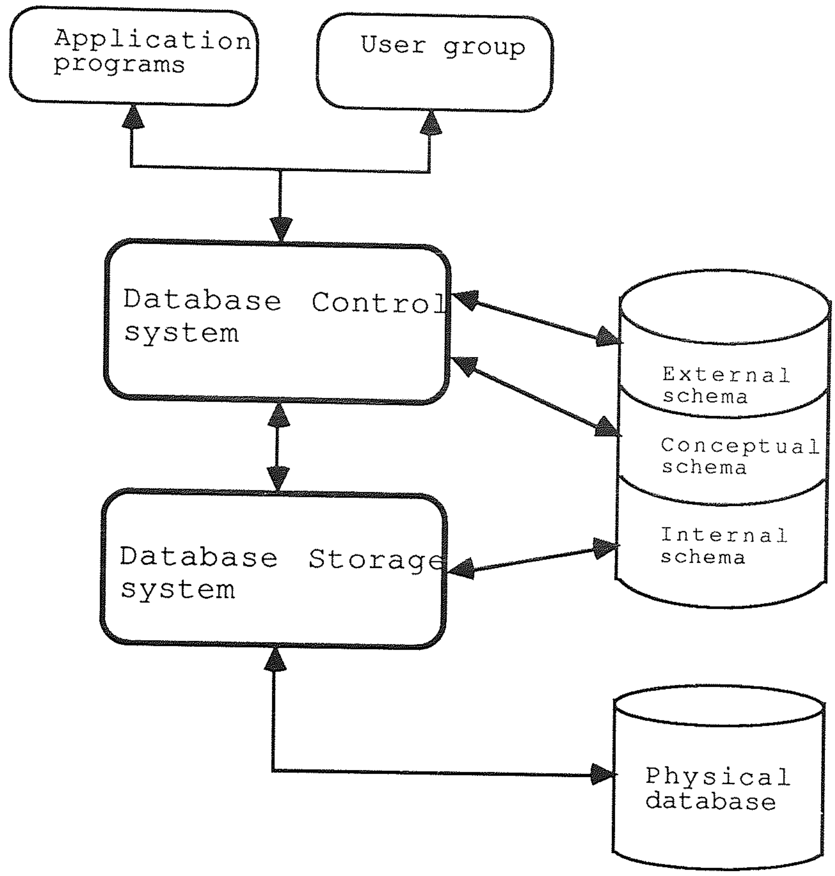


Fig 5.4 Showing the two main operating modules of of a typical Database Management System

shareability of data by many users. Thus, the primary function of any DBMS is to provide multiple user views and allow users to store, retrieve and update the data efficiently.

In addition, comprehensive DBMSs provide a number of other important functions [36]. These may include the following:-

(a) Provision of a data Dictionary/Directory (see Sect 5.3.2)

(b) Ensuring the transaction integrity i.e. ensure that data is committed to the database only if the transaction has been successfully completed.

(c) Recovery of the database to a known state in the event of a system failure.

(d) Concurrency control in multi-tasking or multi-user environments.

(e) Data communication interfacing to allow transactions from remote terminals.

(f) Integrity services to assist the user and database adminstrator in maintaining data integrity.

Contemporary DBMS products differ both in the number of functions provided and in the manner in which the functions are performed. In the more sophisticated products available for mini and mainframe computers, many of the functions are performed more or less automatically while in others, users have to effect some of the functions themselves using facilities provided by the system. Most DBMSs available on microcomputers fall into the latter category.

(3) **Language interfaces**

The facilities for defining and accessing a database are provided by the DBMS through various language interfaces. Two major categories of these may be identified: Data

Definition Languages (DDL) used mostly by the database adminstrator to define the schema and subschemas, and Data Manipulation Languages (DML) used mostly by programmers to manipulate data.

DMLs may be categorized further into two: Host language DMLs and self-contained DMLs. Host language DMLs are used to transfer data between the database and application programs and are essentially extensions of the host language such as FORTRAN or PASCAL. Self-contained DMLs are, on the other hand, special to each DBMS and cannot be embedded in a host language: in this case, the transfer of data between the database and a program in a high level language is not direct. An example of the former is the 'INGRES' relational system which can be used with nearly all the common high level languages. An example of the latter is the 'dBASE II' relational system which is available for most micros using the CP/M or MSDOS operating systems. This was the DBMS used in this research work and is therefore described in more detail in Appendix 1.

There are currently many DBMS products available on the market. The selection of a particular system has to be based on the relative importance of the functions enumerated in Sect 5.3.2 to the user organization, the language interfaces required as well as the existing computer systems in the establishment.

5.4 **Database design**

Database design is the process of developing database structures (corresponding to the three views of data shown in Fig 5.2) from user requirements. The basic objectives are to enable users to obtain the exact data they need to perform their duties within the organization and to make the data available in a reasonable amount of time.

Teorey and **Fry** [37] have developed a general model for database design consisting of four major steps as shown in Fig 5.5. The process, pursued in the present work, is outlined in the subsections that follow.

5.4.1 Requirements formulation and analysis

This activity, also termed 'functional analysis' [38] involves the analysis of the functions in the organization and the information requirements associated with them. In systems engineering terms, this activity constitutes the formulation of design specifications for the database structure and should therefore cover all conceivable requirements of the organization with reference to the database structure. In this regard, the analysis of generalized information transformation requirements for the product-realization methodology presented in Sect 3.6, if interpreted in terms of a specific manufacturing environment,

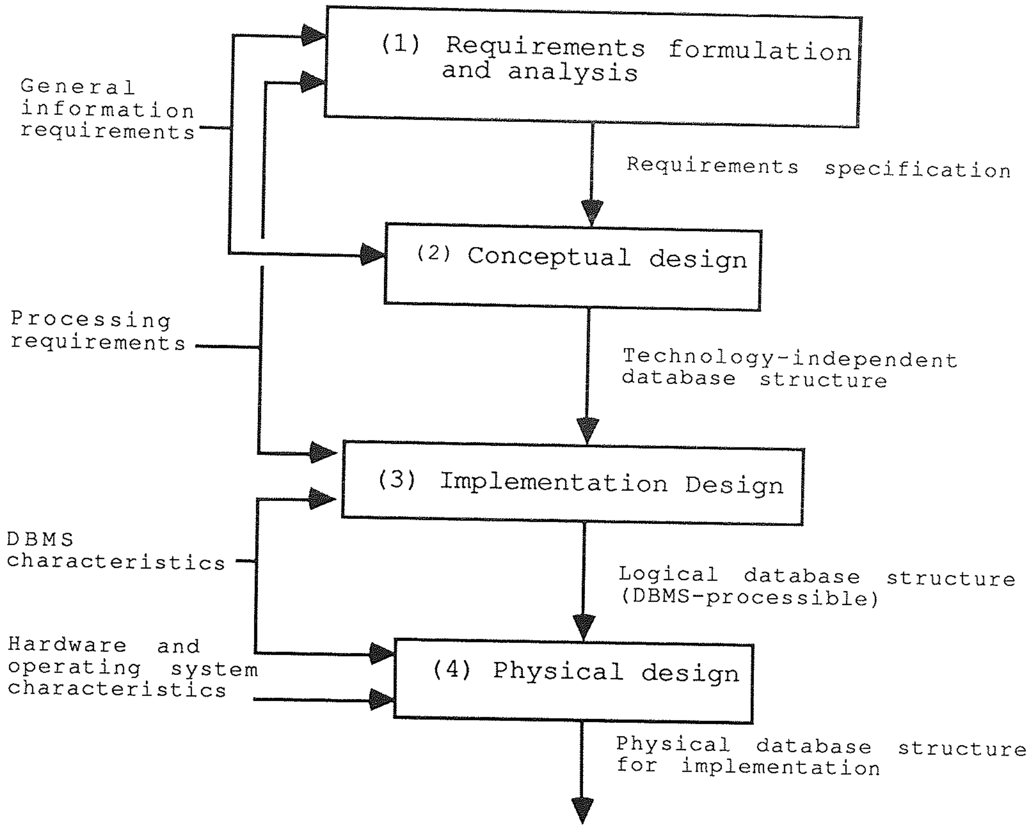


Fig 5.5 Showing the major steps in database design

would provide a solid foundation for the data analysis activity in that environment.

5.4.2 **Conceptual Design**

During conceptual design, the data elements of an organization are identified, analyzed and organized to form the conceptual view (see Fig 5.2), which is a high-level representation of the requirements identified in the functional analysis stage. The objective is to produce a flexible structure which can cope with future changes in the data management requirements of the organization.

The different methods for conceptual design proposed in the literature generally fall in one of two categories. The first is a top-down approach where it is assumed that the database designer is an expert in the database application domain and that he is therefore able to identify the data of interest and synthesize the conceptual model. The second is a bottom-up approach where individual data views are formulated first and then integrated into a single global view. Examples of methods belonging to the former and latter categories respectively are the 'Entity-Relationship approach' and the 'Data Collection approach' cited by **Chaudhuri** [38] and **Avison** [39]. For reasons given later in Chapter 6, the Entity-Relationship (E-R) approach was used in the research work as is therefore introduced below.

5.4.2.1 **The Entity-Relationship approach**

According to **Chaudhuri** [38], this is the most widely used technique for data analysis. It begins by identifying the entities of interest in the database application domain. Using the results of functional analysis, relationships of interest among the entities are identified and displayed in an E-R diagram such as Fig 5.6. Further relationships of interest can then be identified: these could be either of the entity-to-relationship or relationship-to-relationship type.

Fig 5.6 Showing a partial Entity-Relationship diagram for a database system for gearbox design

In Fig 5.6, BEARING and SHAFT are entities and CARRIES is a many-to-one relationship between them i.e. a bearing can carry only one shaft while a shaft can be supported by more than one bearing. The E-R diagram is very useful both for design as well as subsequent database maintenance because it is a snapshot of the database structure.

The Entity-Relationship approach is popular probably because it only identifies the essentials at the first step and therefore does not burden the designer with unnecessary detail. Only when a satisfactory overall structure of the conceptual model has been synthesized are the detailed attributes of the entities and relationships added. In the example above, the attributes of SHAFT can include its various diameters, material name etc., while those for BEARING can include bore diameter, catalogue number etc.. An attribute of the relationship CARRIES is the 'required geometric fit' between a shaft and a bearing.

5.4.3 **Implementation design**

The structure that results from conceptual design is a model of the organization's data, completely independent of any DBMS, or any other software or hardware considerations. On the other hand, every contemporary DBMS is designed to work with a database that has a particular logical structure. The purpose of implementation design is therefore to map the conceptual data model into an internal model (or 'schema') which can be processed by a particular DBMS and satisfy the full range of requirements from integrity and consistency constraints to efficiency for projected database growth. As shown in Fig 5.5, the main inputs to the process are the conceptual model (such as the E-R model), processing requirements (obtained from 'functional analysis') and the characteristics of the particular DBMS chosen.

The logical database models developed so far fall into two major groups: 'relational' and 'non-relational'. the main difference between the two groups is in the way that relationships among data entities are expressed. This is illustrated below with a specific example.

Fig 5.7(a) shows the record schemes that can be developed for the entities shown in Fig 5.6. The records in the two schemes must clearly be related. This relationship can be expressed in two different ways as illustrated in Figs 5.7(b) and 5.7(c).

SHAFT

Shaft No	Material name	
1	mild steel	
2	m/s, norm	

BEARING

Bearing designation	Bore dia	
22208C	40	
22209C	45	
22210C	50	

(a) Showing record schemes for two entities

SHAFT

Shaft No	Material name	
1	mild steel	
2	m/s, norm	

BEARING

Bearing designation	Bore dia	
22208C	40	
22209C	45	
22210C	50	

(b) Showing an EXPLICIT way of expressing relationships between records in the two record schemes

SHAFT

Shaft No	Material name	
1	mild steel	
2	m/s, norm	

BEARING

Bearing designation	Bore dia	
22208C	40	
22209C	45	
22210C	50	

CARRIES

Shaft No	Bearing designation
1	22209C
1	22210C
2	22208C

(c) Showing an IMPLICIT way of expressing relationships between records in the two record schemes

Fig 5.7 Showing two prevalent methods of expressing relationships in logical data models

Fig 5.7(b) illustrates the method used in non-relational models whereby the relationships are expressed 'explicitly', generally by keeping pointers of associated records as part of the record. The main non-relational models are the 'network' and 'hierarchical' models mostly based on the proposals of the CODASYL DBTG [**40**], and which were prevalent in commercial systems until the early eighties. These models are covered extensively by almost all older publications on database technology.

Fig 5.7(c) illustrates the method used in the relational systems more recently advocated whereby the relationships are expressed 'implicitly' by the existence of a common attribute (such as 'Bearing designation' in the example) in the related record schemes.

The relational approach was proposed by **Codd** [**41**] in the early seventies and was aimed at avoiding many of the drawbacks of the non-relational systems in use. It is claimed to provide advantages in data independence, ease of use and flexibility [**42**],[**43**]. for these reasons, a relational DBMS (DBASE II) was chosen for the research work. The relational approach is therefore discussed in more detail hereafter.

5.4.3.1 **The relational data model**

The relational data model provides levels of data consistency and independence which are difficult to achieve with non-relational models: it is therefore seen by many as the model of the future. The main concepts underlying the model are outlined in the subsections (1) to (4) that follow.

(1) **Definitions**

The mathematical concept underlying the relational model is the set-theoretic 'relation' which is a subset of the Cartesian product of a list of domains [**42**]. A domain is simply a list of values. The members of a relation are called tuples. Fig 5.8 shows three examples of relations.

SHAFT

Shaft No	Material name	Length
1	mild steel	100
2	m/s, norm	155

BEARING

Bearing designation	Bore dia	Outer dia
22208C	40	70
22209C	45	75
22210C	50	80

CARRIES

Shaft No	Bearing designation
1	22209C
1	22210C
2	22208C

Fig 5.8 Showing three examples of relations

It helps to view a relation as a two-dimensional matrix where each row is a tuple, each column is a domain and which has the following properties:-

- Each column contains values about the same attribute andeach matrix cell value must be single-valued.
- Each domain (column) has a distinct name.
- Each tuple (row) is distinct; duplicate rows are not allowed.
- The ordering of the tuples is immaterial semantically.

A short-hand notation can be used to abstractly represent the relations shown above as follows:-

SHAFT (Shaft:no, material:name,length)
BEARING (Catalogue:no, bore:dia, outer:dia)
CARRIES (Shaft:no,Catalogue:no)

In this form, that attribute (or attributes in combination) for which no more than one tuple may have the same (combined) value is called the 'primary key' for the relation. In the example relation schemes above, the primary keya are (shaft:no) and (Shaft:no,Catalogue:no) for the relations SHAFT and CARRIES respectively.

(2) **Relationships in the relational model**

In Fig 5.8, SHAFT and BEARING are primary data groups that can exist independently from all other data. The basic construct for representing a relationship in the relational model is to form a new relation such as

CARRIES, whose primary key is a combination of the primary keys of all the relations involved in the relationship. The new relation thus formed can have other non-prime attributes (such as 'type of fit required' for CARRIES relation) attached to them. The new relation can also take part in higher level relationships with other entities and relationships in the logical model.

(3) **Functional dependency and normalization.**

Any table that satisfies the five properties listed in (1) is a relation. However, experience has shown that such relations can still possess some undesirable properties with regard to database maintenance. In general, inconsistencies can occur in the database after records have been inserted, deleted or modified. The process of ridding a database definition of these possible anomalies is called 'normalization' [**42**].

Central to the process of normalization is the concept of 'functional dependencies' among the attributes of a relation. A functional dependency (FD) is defined as follows:-

Attributes B are functionally dependent on attributes A, or

$$FD(A_1,A_2,A_3,\ldots,A_j) = B_1,B_2,B_3,\ldots,B_k$$

if the value B_i in any tuple is always fully determined by the set of values $A_1, A_2, A_3, \ldots, A_j$

To avoid anomalies in the database, a general requirement recognized is that all dependencies within a tuple of a fully normalized relation should be functional dependencies between each of the non-prime attributes and the prime key.

A number of different properties or 'normal forms' for relation schemes with dependencies have been defined [**41,42**]. These normal forms guarantee that most of the anomalies referred to above do not occur. The most significant of these is the 'third normal form' (3NF) which is considered by many to be adequate for most normal database implementations. The basic steps in the normalization process to 3NF are shown in Fig 5.9 and an example of the process is given in Fig 5.10.

Further normal forms namely 4NF and 5NF have been defined for further refinement of the logical model but, according to **Cardenas** [**42**], their payoffs in actual database practice still remain to be proved. These are therefore not introduced here.
It is important to stress that the definitions of functional dependencies and therefore, the properties of the resulting normalized relations are dependent

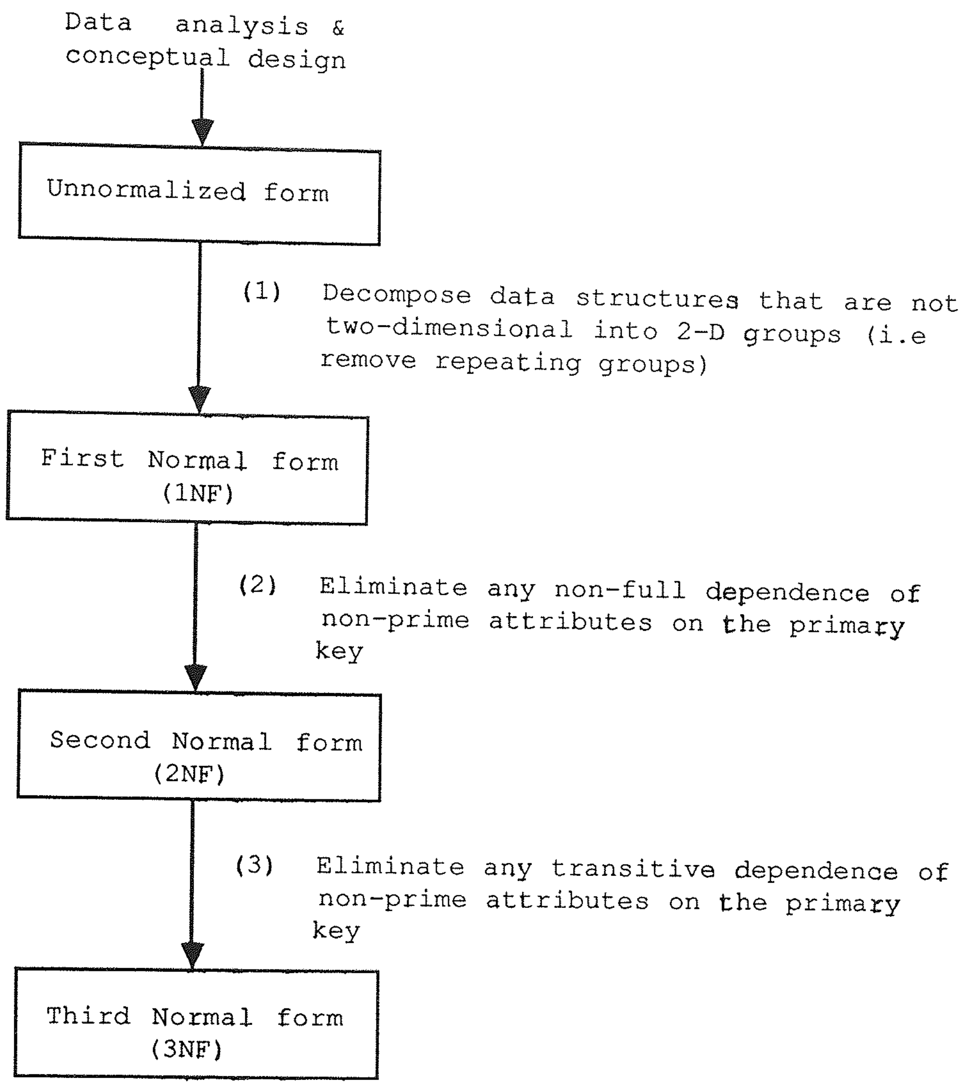
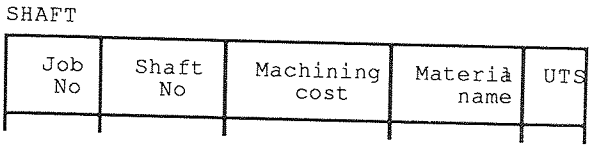
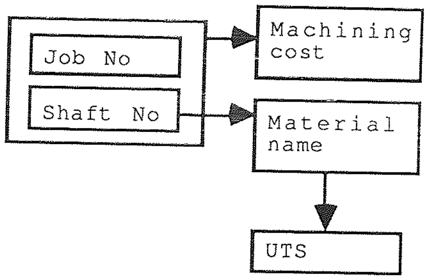


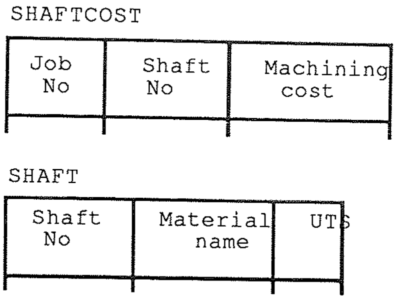
Fig 5.9 Showing the main steps in the normalization process to third normal form

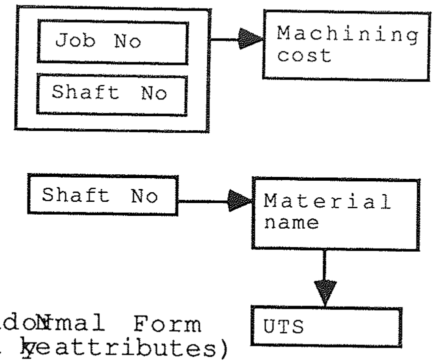
solely on the semantics of the database as percieved by its designer. The FDs are derived by reasoning about the data; they are not deducible from any collection of data. This emphasizes the importance of the data dictionary module in the database system.



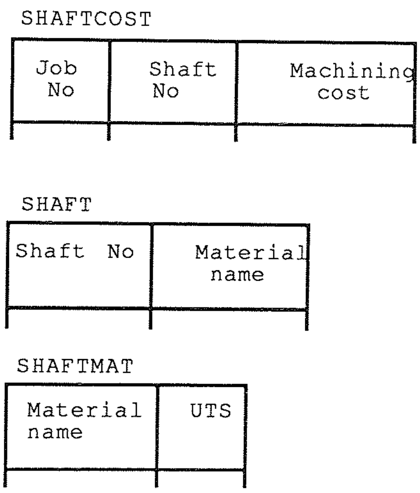


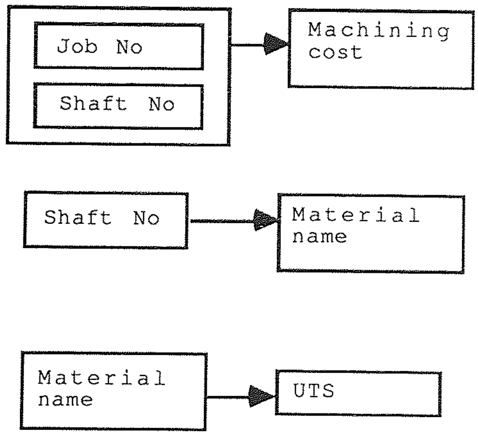
(a) Database in First Normal Form
(i.e. no repeating groups of data)





(b) Database structure in Second Normal Form
(No partial dependencies on key attributes)





(c) Database structure in Third Normal form
(No transitive dependencies on key attributes)

Fig 5.10 Illustrating the normalization process with an example

(4) **Data manipulation in the relational model**

Different users of the same database will percieve different sets of data and data relationships i.e. different relations. Relational DBMSs therefore provide the facilities, for example, to extract a set of domains to create relations of smaller degrees or join sets of domains to create relations of larger degrees, in order to match a particular user's perception of the database. In effect, such facilities allow each user to view the database as a time-varying collection of relations of assorted degrees. It is these 'cutting and pasting' operations that give relational database systems a degree of flexibility that is difficult to achieve in non-relational systems.

The relational data model is based on the well-defined theory of relations and sets. Therefore, the data manipulation languages (DMLs) implemented in contemporary DBMSs also have the same mathematical foundation [**43**]. In practice however, the user of a proprietary DBMS is able to accrue the benefits of the sound theoretical foundation without being burdened with the mathematics involved in the manipution of relations in the model. Two classes of special-purpose DMLs characterize the methods presently available for manipulating data in relational databases: relational algebra and relational calculus.

(a) With DMLs based on relational algebra, the user manipulates the relations in the data model in a procedural fashion by using certain defined operators to derive what he wants. Relational algebra manipulates one or two relations as operands and produces a new relation as the result. the basic operations include the four main ones prevalent in set theory i.e. UNION, INTERSECTION, DIFFERENCE and CARTESIAN PRODUCT. The first three are applicable to two or more relations which have identical schemes while the last one creates a set of all combinations of arbitrary relations.

Three other operations have been specifically defined to deal with relations namely SELECTION, PROJECTION and JOIN.

- The SELECTION operation retrieves all tuples of a relation that satisfy a certain condition and constructs a new relation containing only the selected tuples.
- The PROJECTION operation extracts specified domains from a relation and forms a new relation containing only the specified domains.
- The JOIN operation does the opposite of the PROJECTION operation. It takes two relations that have a common domain and joins them to form a new relation.

(b) A relational calculus is a mathematically-oriented notation for defining a relation to be derived from existing relations in the database. It is non-procedural in that it permits the user to indicate what he wants to obtain without having to specify many details of how to obtain it. It is left to the DBMS to determine what relational algebra steps or equivalent operations are necessary to derive the required relations.

5.4.4 **Physical database design**

Physical database design is the process of developing an efficient, implementable database structure. It is concerned with how the data is stored on physical devices rather than how it appears to the user.

The main inputs to the physical design process are shown in Fig 5.5. The options open to the designer as regards the optimization of the physical structure are mostly determined by the data description language facilities provided by the particular DBMSn use. Physical design may include the design of stored record formats, record clustering (i.e. physical grouping of records to minimize access times), and the design of the access methods (i.e. techniques for storing and retrieving records). Several trade-offs among access times, storage requirements and data redundancy must typically be made at this stage. Once physical design is completed, the database is ready for implementation.
The design of a database structure is a complex task: it therefore requires an organized approach such as the one outlined in the preceeding sections. It is also an iterative process, often requiring trial and error. In practice, it

requires the commitment and participation of the whole organization: the role of the database adminstrator/designer in a database environment is therefore a very crucial one.

5.5 Concluding remarks

A brief presentation of the fundamentals of current database technology and practice has been given in this chapter in order to justify their consideration for application in the research work. The essential feature in any database system has been shown to be an integrated collection of data which is structured on natural data relationships so that it provides all necessary access paths to each unit of data in order to fulfill different user requirements. The main benefit of the approach is data shareability which is essential for integration in computer aided systems.

The presentation given in the chapter has been from the point of view of a potential user of database technology to fulfill the data management requirements in a specific application area i.e. in CAE systems. In particular, the use of a proprietary DBMS removed the need to describe at length the more theoretical aspects of the technology.

The database approach as outlined in the chapter has been successfully applied in commercial systems. Its applicability in CAE systems is currently an emerging research topic. The application of the approach in engineering systems is presented in chapter 6.

C H A P T E R S I X

APPLICATION OF THE DATABASE CONCEPT TO THE MANAGEMENT OF INFORMATION IN COMPUTER AIDED ENGINEERING SYSTEMS

6.1 Introduction

The central theme of this thesis is the achievement of integration in CAE systems. The working notions regarding an integrated system have been presented in Chapter 4. In Chapter 5, an approach to information management which seems to offer the flexibility and generality required for integration i.e. the database approach, has been identified. The main objective in this chapter is therefore to provide the basis on which the database approach may be applied to the management of information in CAE systems preparatory to the detailed description given in Chapter 7 of the development of an integrated CAE system.

Following a brief review on databases in engineering, a design specification for an engineering database system is presented, drawn up after examining three important issues namely (a) the nature of discrete mechanical products, (b) characteristics of engineering information and (c) the nature of engineering activities. A database system based on a multi-layered relational database schema is then shown to be a viable approach to CAE system integration.

6.2 Databases in engineering - a brief overview

In Chapter 3, the data used in engineering activities has been classified into three main groups i.e. engineering, product-modelling and information adminstration data. The current application of database concepts in engineering is reviewed hereafter in terms of these data groupings.

6.2.1 Engineering databases

Engineering data has been defined to be registers of industrial standards and know-how that are valid beyond a specific product. Common examples are materials data, catalogued data from suppliers of bought-out machine elements and publications from institutions such as the British Standards Institution.

Engineering databases are characterized by a very high volume of data, modelled using very few well-defined data entities. Furthermore, both the structures and the data itself are basically static in nature i.e. engineering databases are mostly referenced rather than updated. Once the database structure has been established and the data has been entered, the main concern thereafter is minimizing the access time to data.

The characteristics of engineering data as outlined above are essentially the same as those of commercial data.

Conventional database techniques can therefore be successfully applied to the management of engineering data. In particular, the efficiency associated with non-relational systems makes them particularly suitable for handling such data.

6.2.2 **Product-modelling and information-adminstration databases**

Product-modelling data (PMD) has been defined as that data which gives an at-all-times correct description of the product under development. Information-adminstration data (IAD) is used to keep track of both engineering data and product-modelling data. Both the types and quantities of PMD and IAD will change as the product realization process continues. More significantly, the structures of the data entities modelling PMD and IAD may also need to be altered as the process continues in order to maintain data integrity. It is mainly this dynamism, compounded by the iterative nature of the product evolution process which brings into question the adequacy of conventional database practices and technology for managing product-modelling and information-adminstration data. An important part of this research was thus directed towards determining the level of adequacy and developing appropriate new techniques to extend the concepts to CAE applications.

The dynamism of PMD and IAD is most pronounced in the design phase of the product realization process. Not surprisingly therefore, research in databases in CAE has concentrated on the problems of handling CAD data, and geometric data in particular. Two research trends are apparent in the literature as follows:-

(a) Research aimed at developing methodologies for handling CAD data using the conventional data models and DBMSs [**44,45,46**] or extensions of these [**47**,48]. **Katz** [**49,50**], **Gray** [**51**] and **Eberlein** & **Wedekind** [**44**] provide extended general discussions of the drawbacks of conventional database tools with regard to the management of CAD data. The common view is that no existing commercially-available system supports the complete range of facilities needed to support design activities. At the conceptual level, the problems cited include the inability to define data structures dynamically, and the difficulty of organizing a design hierarchy or supporting multiple design representations. At the implementation level, the main drawbacks claimed are that contemporary database systems are difficult to use and suffer from poor performance, that extensions to databases are disruptive and costly, and that interfacing to a database requires too much work.

Generally, the latter two problems mentioned in the last paragraph are less significant in database systems based on the relational data model (see Sect 5.4.3.1) than in those based on other models. It was shown in the course of this research work that these problems can be alleviated even further by the creation of a data directory/dictionary system specially geared towards assisting the design and maintenance of a database. The system created as part of the research is described in Appendix 4.

(b) Research aimed at developing new design methodologies and models for product modelling databases [**52,53,54,55**] including those synthesized from existing models. Although the originators of these new approaches have claimed their adequacy for handling PMD, it is difficult to comment on their general applicability because most of these are presented in the literature as specific CAE products, with limited details regarding their theoretical foundations.

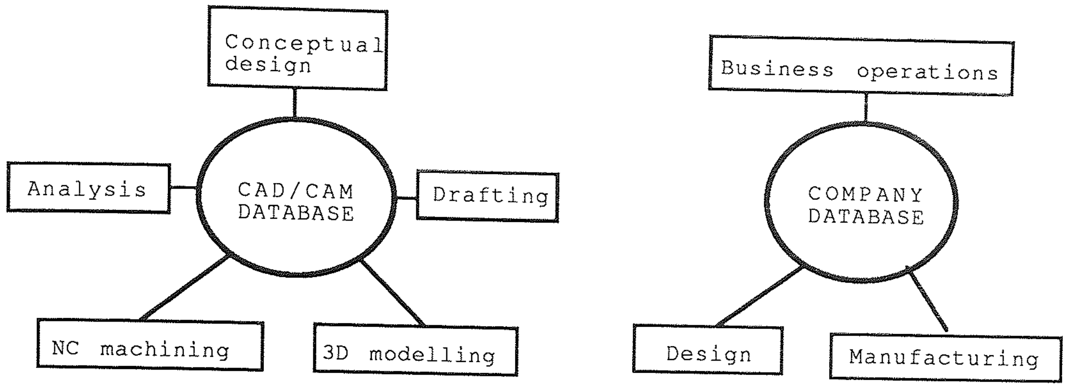
With the growth in popularity of the concept of computer-integrated manufacture, a main research interest is now in extending the database concept to the management of the whole information resource of the company.

6.3 **Data modelling in an integrated environment**

The brief review of databases in engineering given in Sect 6.2 shows that so far, the general objective has been to use database technology to enhance the performance of engineering activities within an organization. Even where the term 'integration' has been used in the literature, it has been in the narrow sense of facilitating the sharing of data among various engineering activity modules. Although this is perhaps just a stage in the natural progression in the technology, it can lead to the creation of an engineering 'island of automation' as discussed in Chapter 4. In contrast, the objective of integration has been defined in Chapter 4, to be the optimization of the operation of the total business over and above that of individual functions in it. With regard to information management, the requirement is therefore that data used in design, in production and in business operations is organized in an integrated manner so that, for example, feedback from production and sales can easily be obtained and used to effect a better design of the product. The difference in the objectives is depicted in Fig 6.1.

6.3.1 **The need for a common data model**

The heterogeneity of data that must be managed in an integrated system raises one major requirement with regard to the application of the database approach in such systems. It is that a common data model is required so as to enable data



(a) CAD/CAM integration

(b) Business integration

Fig 6.1 Showing two different objectives of current research in integrated CAE systems

sharing among the different areas of activity. Thus the problem is no longer how best to model design, production or commercial information but is how best to organize them all collectively. This requirement is depicted in Fig 6.2. The company database shown in the figure needs to have a particular logical structure. In practical terms today, this also implies that it has to come under the control of a particular DBMS. This does not however exclude the use of distributed database concepts which permit centralized management of data but at the same time allow for the geographical distribution of today's organizations. The design of the contemporary data models as well as the proprietary DBMSs based on them, was motivated by the need to manage large volumes of commercial data. However, this cannot necessarily preclude their utilization in the management of other types of data such as engineering data. Furthermore, current literature is inconclusive as regards the adequacy of conventional database techniques for engineering information

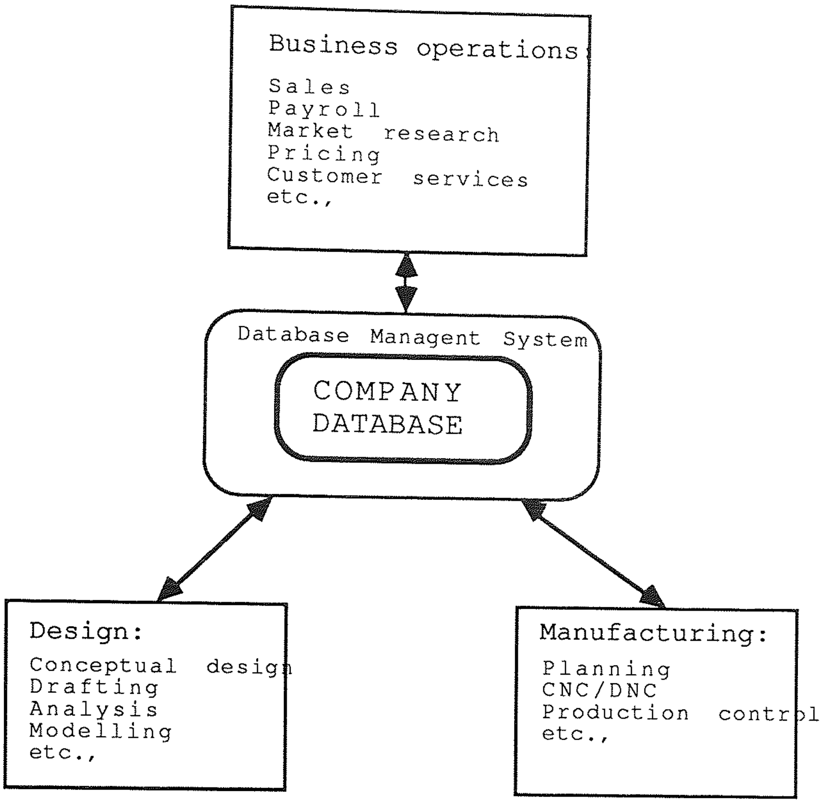


Fig 6.2 Depicting the requirement for a common data model in an integrated manufacturing system

management. With this in mind and and with the requirement for a common data model in an integrated environment, the thrust of this research work was directed towards developing model formulations which would enable database tools to be applied to the new problem of CAE system integration: an evaluation of this approach would then indicate the desirability, or otherwise, of developing completely new tools.

6.4 **Requirements analysis and specification for an integrated CAE database system**

The focus on the integration of the whole business environment requires a different specification for the database system from those that appear in the literature for commercial or engineering design systems. Since some of the ideal data management requirements in the individual activity areas such as design or business operation can be very different, the specification required has necessarily to represent a compromise among these.

It has been shown in Fig 5.5 that when setting up specifications for a database system, the two main issues that should be examined are (a) the nature of the data belonging to the "universe of discourse" i.e. the boundaries of the database system, and (b) the data usage envisaged, deducable from the nature of the activities that are going to use the data. These issues are examined in Sects 6.4.1 and 6.4.2 respectively for an integrated CAE database system. The observations made apply mostly to product-modelling and information adminstration data. In addition, since the research work was particularly concerned with the realization process for mechanical products, another relevant issue is the nature of discrete mechanical products. This is examined in Sect 6.4.3.

6.4.1 **The nature of CAE data**

The data found in a manufacturing environment has a number of semantic and structural properties that need to be taken into account in the design of an appropriate database system. The more significant of these are as follows:-

(a) Its structure is complex: many relationships exist among relatively few data entities. Furthermore, these relationships may be of the many-to-many type and may themselves constitute higher-level data entities, possessing independent attributes of their own.

Consider the design of a pair of meshing gears. An obvious data entity of interest is the individual gear, with attributes such as the gear material, pitch circle diameter etc.,. However, other important design parameters such as gear ratio, centre distance, power etc., can be expressed only as attributes of a relationship between one gear and another. This is demonstrated in Fig 6.3.

Apart from entity-to-entity connections, it may be necessary to form entity-to-relationship and even relationship-to-relationship connections. (See Fig 7. for examples). This necessary hierarchy of relationships greatly complicates the maintenance of data integrity in a CAE system database.

(b) Many of the relationships expressed in engineering databases often model hierarchies with a varying number of levels. In the modelling of shaft data for

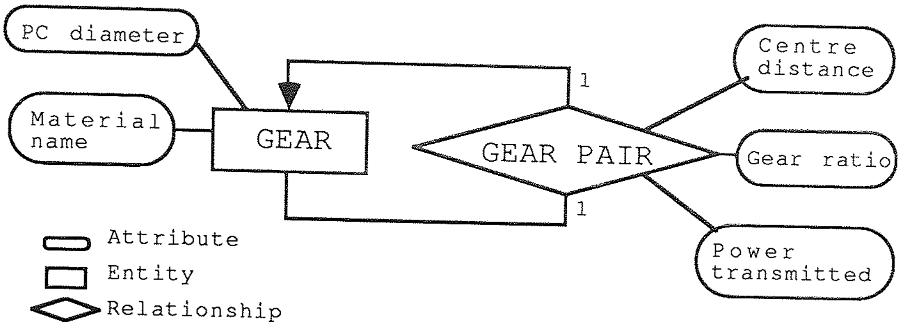


Fig 6.3 Showing a partial Entity-Relationship diagram for a database structure for gear design

example, it may be necessary to store the position of a keyway. However, the keyway itself has to be described at a lower level in terms of its length, depth etc.,. This recursive property has to be captured by the data model.

(c) In design and manufacturing, the order in which activities take place is extremely important since it can represent the temporal or procedural relationships of the data objects.

Consider the design of a shaft for a gearbox. Two possible procedures are shown in Fig 6.4. In both cases, the data that describes the shafts is continually changing. Furthermore, the optimum data structure required to support procedure (1) may be quite different from that for procedure (2).

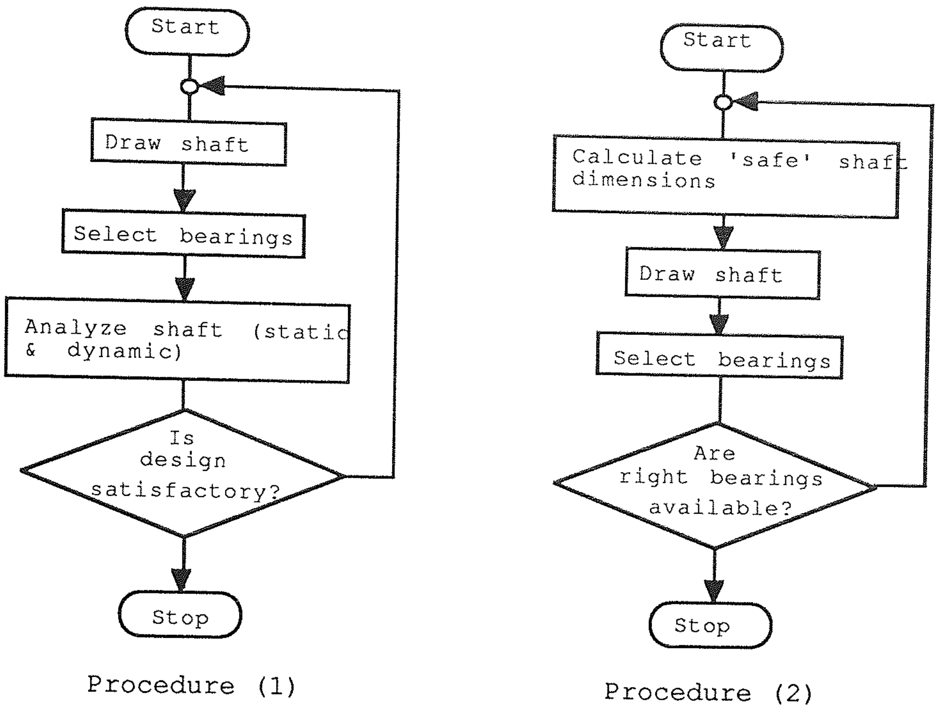


Fig 6.4 Showing two possible procedures for shaft design

(d) Engineering designs evolve over time. New versions of system components are created and incorporated into new configurations of the design. It is normally necessary to store a number of versions of the same design.

(e) The CAD/CAM application area in particular uses a number of complex data types such as vectors, matrices, sets etc., in the storage and manipulation of data. On the other hand, contemporary database management systems have relatively limited datatyping capabilities, recognizing only a few basic data types such as integers, real numbers, characters and Booleans. Mapping between the complex CAD/CAM data types and the relatively those supported by the DBMS may introduce extra processing requirements, depending on the sophistication of the DBMS in use.

The significance of each of the data properties listed above will vary, depending on the type of engineering activities obtaining in a particular engineering enterprise. In general, their significance will be greater in an innovative environment where new concepts and methods are constantly being adopted than in a specialized and established one where nearly all data management requirements are well known.

6.4.2 The characteristics of engineering activities

Engineering activities are generally seen to follow the generic pattern shown in Fig 6.5.

In both manual and non-integrated CAE systems, the acquisition, preparation and verification of input data and the documentation or transmission of output data represents a major part of the total engineering effort. It is in these areas i.e. steps (1) and (5) in Fig 6.5 that integration can provide the most obvious advantages. Tasks (3) and (4) in the figure have received considerable attention as evidenced by the abundance of applications software on the market. As opposed to the short-duration, simple units of work that characterize commercial systems, tasks (3), (4) and (5)

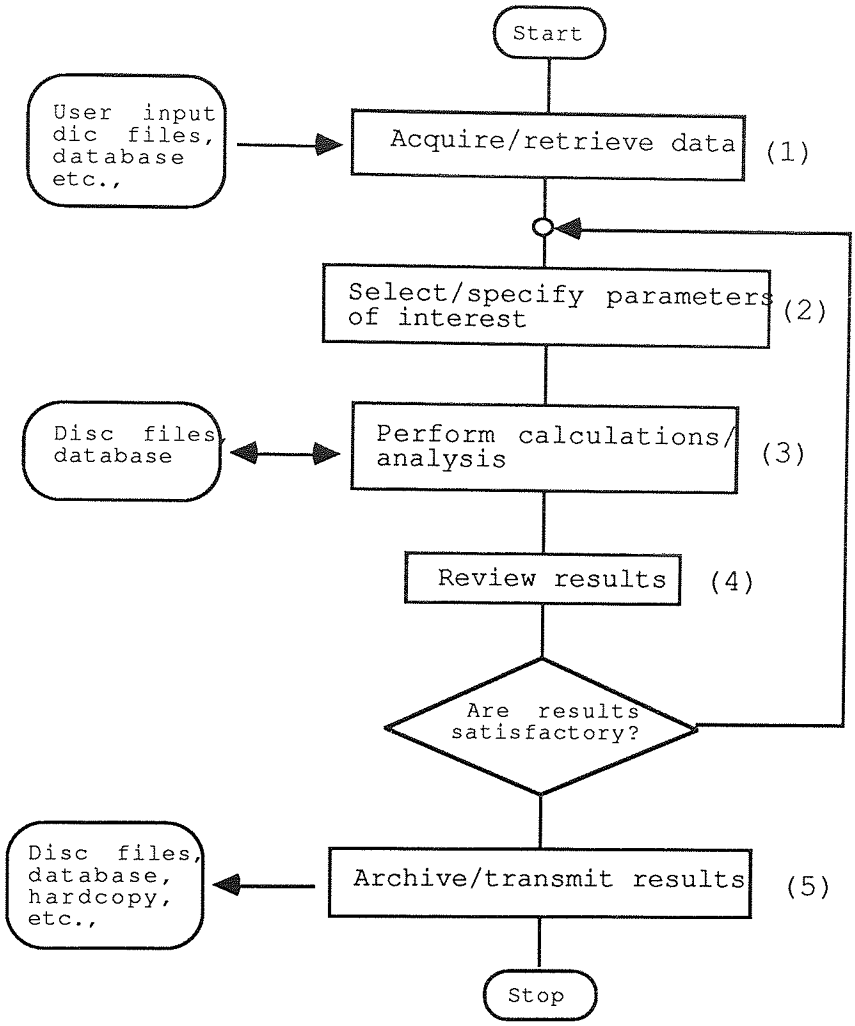


Fig 6.5 Showing the generic pattern for engineering activities

together closely resemble an editing session where the engineer interacts with his data through the set of tools available on the system until he believes that his results are satisfactory enough to be 'released' for subsequent purposes (task (5)). In view of this contrast, the following observations become relevant with regard to a database system for CAE applications:-

(a) The units of data access required in engineering applications are large collections of related records, usually spanning several files. As an example, consider the data requirements of a bearing selection application. These will include the shaft geometry, applied loads and bearing locations. In a relational database for example, these data will span several relations as determined by the conceptual design and normalization of the database structure. Because of the large quantities of data involved, the overhead of entering and leaving the database to extract or modify a record at a time would be too great: it is far more efficient to extract or modify large aggregations of data as a unit.

(b) Engineering applications, and design applications in particular, need not be as closely coupled to the database system as commercial systems are since, by their very nature, most of the run-time is spent in 'number crunching' or interaction with the user rather than in the relatively few database access routines required. Normally, the only database access required 'on-line' is for the referencing of catalogued data for selection purposes and not for the modification of the more intricate product-modelling or information-adminstration data. Thus, the rapid response required in commercial systems database transactions is not as important for engineering applications.

(c) In the design stage, it may be desirable for the database system to have the ability to capture intermediate data i.e. data generated up to the stage where the running of the application program is halted, for whatever reason. This facility may be required because of the following:-

- Since some design applications can be of very long duration, it may be necessary for them to be run in sessions.
- It is normally desirable for other designers to be able to view the in-progress data for purposes of discussion, reporting etc.,. On the other hand, concurrent access with the ability to use or modify incomplete product-modelling data is both undesirable (because of the high sensitivity of its integrity to changes) and unlikely in engineering design.

(d) In an engineering application, a greater part of the necessary data validation work is taken care of by the logic of the program. This reduces the need for validation during the transaction with the database.

(e) Except for purposes of statistical analysis on the contents of the database (for management information purposes), a comprehensive database query facility is not as important in a CAE environment as it is in commercial systems. In fact, direct access to product-modelling data, other than for viewing purposes, is undesirable. It will normally be easier and quicker to examine an object of interest by either browsing through the database in an interactive manner or invoking the application program with which the object was created than to formulate an ad-hoc query using the data manipulation commands provided by the DBMS.

Most of the characteristics of engineering activities listed above can have the effect of making the implementation of a CAE database system easier as regards the interfacing of individual applications to the database structure. This assertion was confirmed by the experience with the experimental CAE system created in the course of this research work, where the main problem area was in the optimization of the database structure rather than in the setting up of an operational system.

6.4.3 **The general structure of mechanical products**

It is generally possible to describe a discrete mechanical product in terms of its assembly, subassemblies and so on, down to the level of individual components. This hierarchical description is depicted in Fig 6.6. The figure also shows that each subassembly and components can be duplicated in different locations in the product.

Consider the structure of a simple industrial gearbox as

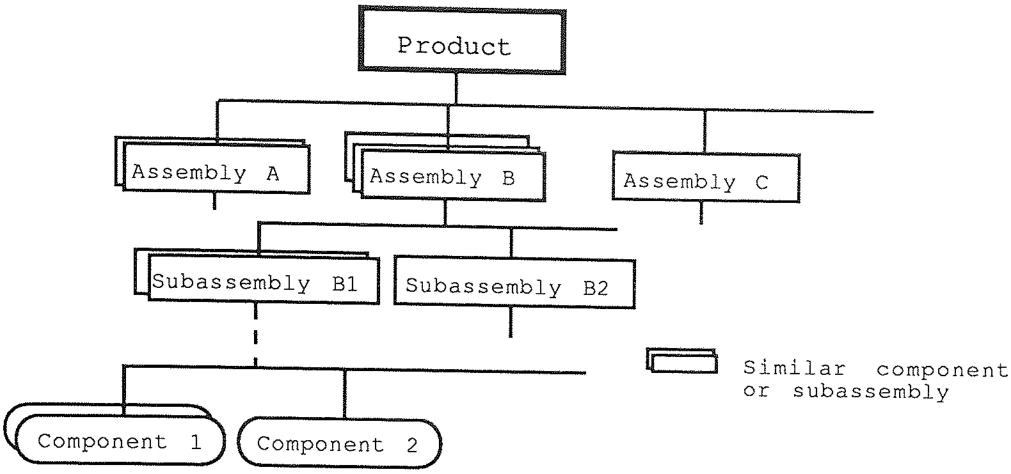


Fig 6.6 Showing the hierarchical description of a a typical discrete mechanical product

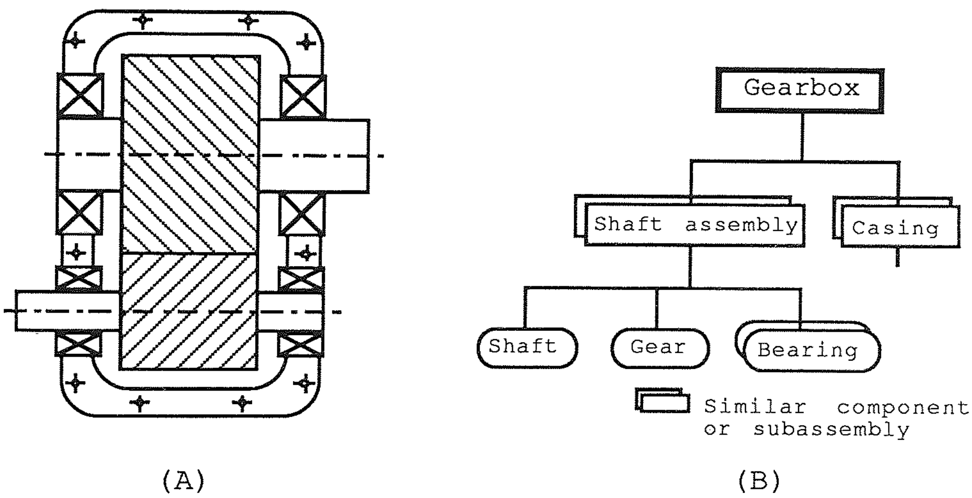


Fig 6.7 Showing a hierarchical description of the structure of a simple industrial gearbox

shown in Fig 6.7(a). At the top description level, the unit can be taken to be made up of two assembly types i.e. shaft assemblies and casing assemblies as shown in Fig 6.7(b). Each shaft assembly can then be described in terms of the shaft itself, the bearings and the gear as the individual components. Similarly, each casing can be described in terms of further lower level components if so defined.

While most of the definitions of subassemblies will be obvious from, for example, the way in which various parts are physically grouped together, they can differ for the same product, depending on the usage envisaged for the product description. For example, for purposes of gear design alone, it may be advantageous to define the two shaft assemblies shown in Fig 6.7 as constituting a higher level sub assembly, since such design always involves two gears.

The ability to describe the structure of a mechanical product in the form a hierarchy provides major advantages with regard to the design of the conceptual database in a CAE system. These are discussed in Sect 6.5.3.

6.4.4 **General specifications for integrated CAE database systems for mechanical products**

A number of general requirements for an integrated CAE database system for the design and manufacture of discrete mechanical products can be drawn from the brief analyses of engineering data, engineering activities and mechanical products given in the previous section and also from the discussion on integrated CAE systems presented in Chapter 4. These are summarized below under two headings i.e. (1) General system requirements and (2) Database structure requirements.

(1) **General system requirements**

These are general requirements with regard to the combination of the database structure, the DBMS and the applications that make up the system. The more important ones are as follows:-

(a) Retention of conventional database manipulation.
A sizable proportion of the data in a CAE system will either be standardized engineering data, completed design data or data used in the business operations of the enterprise. Conventional database manipulation tools can be used to great advantage in the management of such data: these should therefore be retained.

(b) Support for a variety of object representations.
In an integrated system, an object modelled in a database will need to be acted upon by a multiplicity of applications. In addition, each application may require a different representation of the object. The database system must therefore provide a mechanism for multiple representation of objects.

(c) Consistency maintenance across object representations.
The ability to have multiple representation of objects

should be accompanied by a mechanism to maintain consistency across them. If changes are made to a particular representation, the system should recognize the portions in the other representations that could been affected and provide the necessary flagging or, if possible, change these automatically.

(d) Distributed data processing facilities.
An CAE database will generally belong to a distributed computing environment e.g. design workstations in the design office, production management systems in the production department and CNC machines in the workshops. The database system should therefore facilitate access to data from these dispersed computing facilities.

(e) High level language interfaces.
Most of the application programs in a CAE system will be coded in a high level language such as BASIC or FORTRAN. The database system should therefore enable efficient interfacing to a number of such languages.

(2) **Database structure requirements**

These are general requirements regarding the structure of the conceptual view of an integrated CAE system. The two most important are as follows:-

(a) Flexibility

A CAE environment is a dynamic one where new tools and procedures will constantly be introduced as they become available. Therefore the conceptual view may need to be modified or extended frequently in order to accommodate these. The logical data model should therefore offer the necessary flexibility to enable easy and non-desruptive modification.

(b) Hierarchical structure

The schema should reflect the hierarchical structure of discrete mechanical products described in Sect 6.4.3. The explicit expression of 'ownership' of data records in such a structure would ease version control as well as the maintenance of consistency among various representations of the same object.

The specification presented in this section has concentrated on the management of product-modelling and information adminstration data. The assumption has been made that the other types of data found in a manufacturing system can be managed satisfactorily using conventional database tools: the major requirement in an integrated environment is for a database system in which both groups of data can be managed efficiently.

6.5 **An integrated CAE database system**

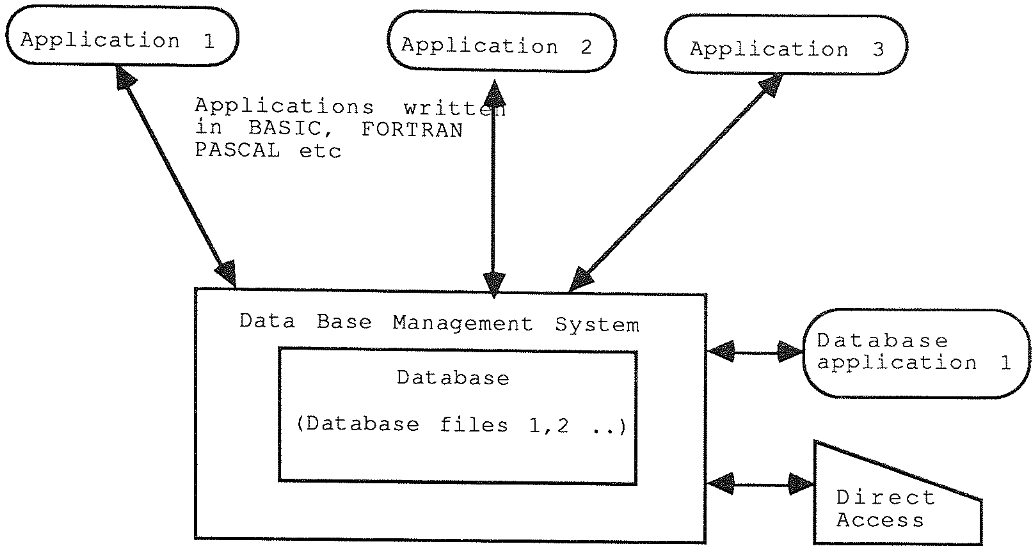
The structure of a proposed CAE database system is described in this section. The overall system concept is outlined firstly, followed by more detailed descriptions of the two novel operational entities in the proposed system namely (a) the system control module and (b) the system database. The implementation of these concepts in a suite of programs concerned with the design of industrial gearboxes will be described in Chapter 7.

6.5.1 **Overall system concept**

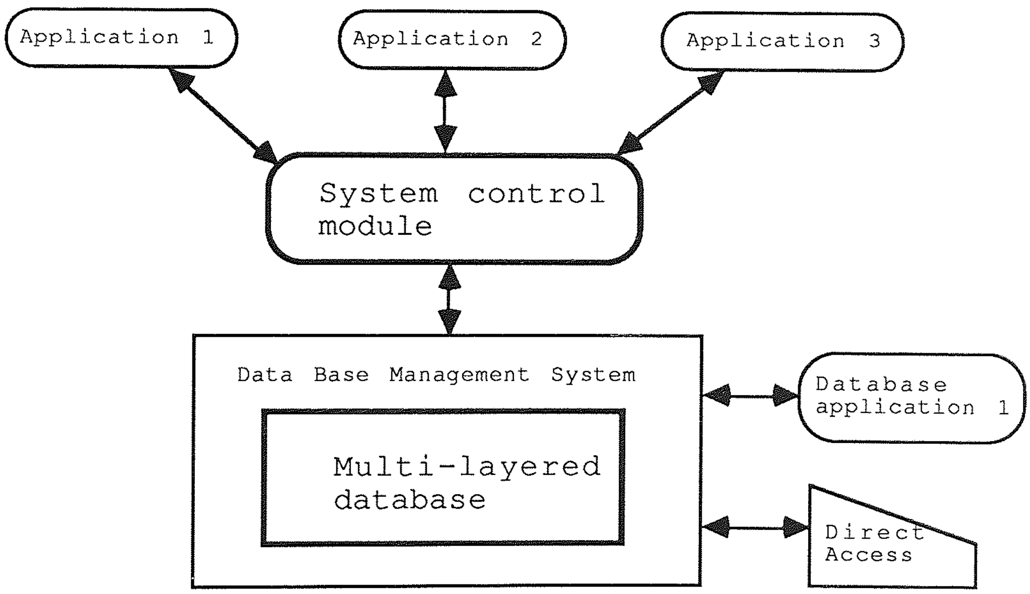
The concept of the proposed system is shown in Fig 6.8(b). The incorporation of a 'control module' and a multi-layered conceptual database structure provides the main difference between it and the conventional database architecture shown in Fig 6.8(a). Further particulars of the new elements are given in Sects 6.5.2 and 6.5.3 while the operational features retained from the conventional database system configuration are given below:-

(a) Application programs coded in high level languages such as BASIC and FORTRAN.

(b) Database utilities in the form of either high level language programs containing embedded database manipulation commands or programs coded in the particular DBMS's own programming language. These utilities will typically be for directory/dictionary queries and for the



(a) Conventional database system concept



(b) Proposed CAE database system concept

Fig 6.8 Contrasting the conventional database system concept with that proposed for integrated CAE systems updating of standardized data such as materials data. Also, some of the more routine queries and updates of production or business data may be effected with such programs.

(c) A direct access facility, principally to allow the database adminstrator to modify or extend the database structure as required.

(d) A database and a proprietary database management system. The DBMSs currently available on the market are based on one of the the three common logical database models i.e. hierarchical, network and relational. Of these, the relational model offers the greatest flexibility of the database structure and has therefore been adopted in the proposed system.

6.5.2 **The CAE control module**

The control module shown in Fig 6.8(b) can be viewed as a common interface program between the database structure and the application programs. Its main function is to regulate the usage of the common data in the database in order to maintain its integrity.

The following scenario will be typical of an established CAE environment regardless of whether a database system is in use or not. A number of jobs (i.e. products under development) will be in progress at the same time. Also, since a typical product is made up from subassemblies and components, a number of different application programs need to be for each job. Furthermore, since the same types of subassemblies and components will generally be duplicated in different locations in the product, a particular application program may need to be run more than once on each job. This scenario is depicted with an example in Fig 6.9.

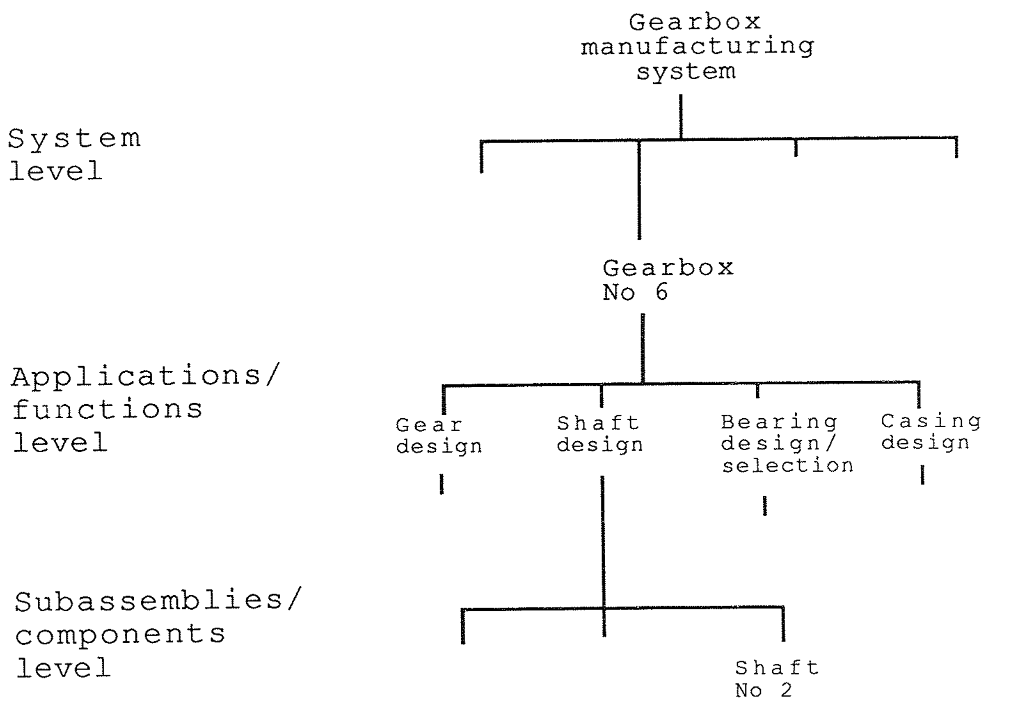


Fig 6.9 Depicting the typical scenario in a manufacturing environment with an example

The hierarchy shown can be duplicated for different products in the same manufacturing system. Also, the last two levels are interchangeable in the sense that a number of different application programs may need to be run for the same component. The role proposed for the control module in such a scenario is described in Fig 6.10.

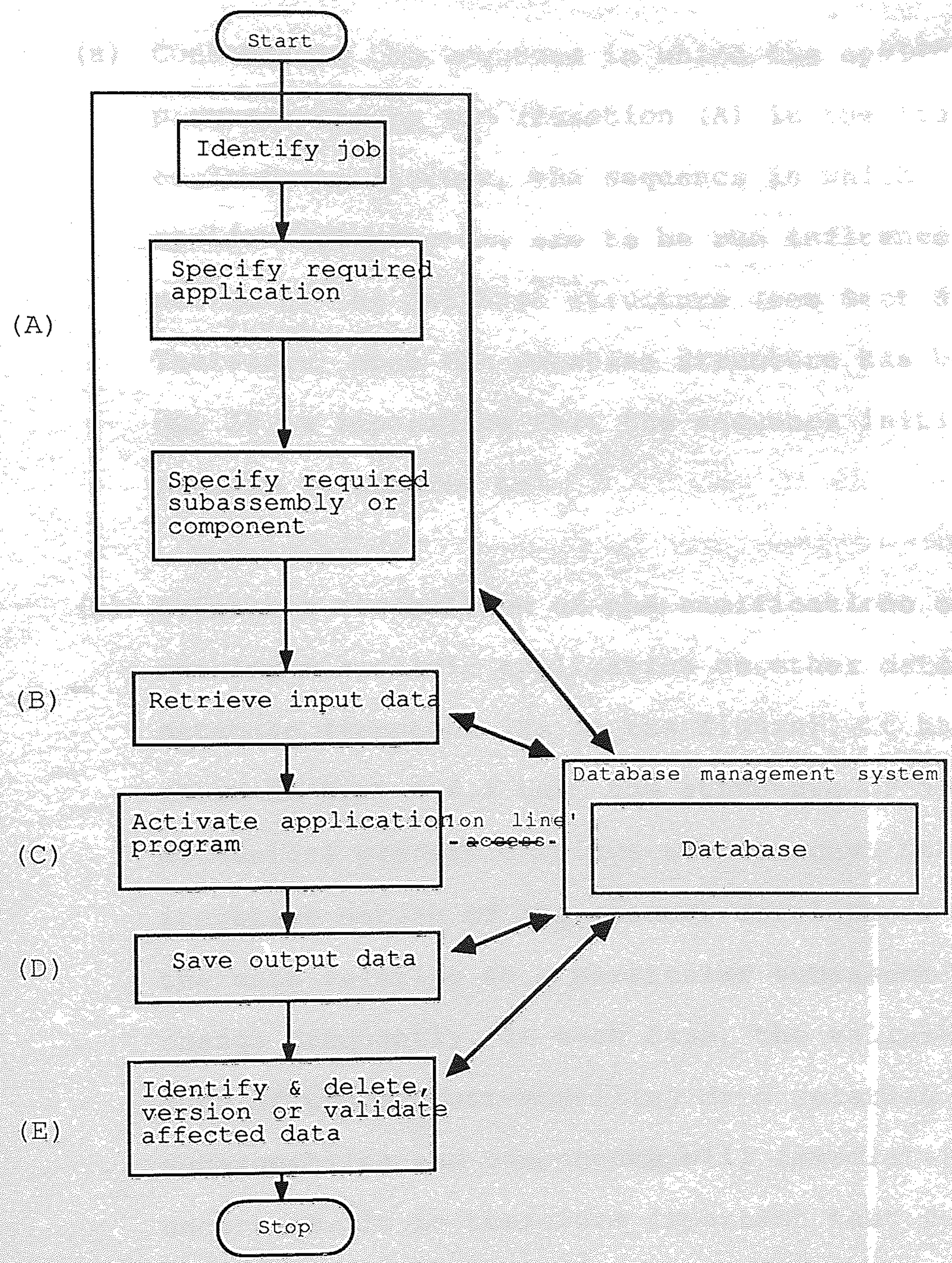


Fig 6.10 Showing the operation of the proposed CAE database system control module

The module maintains data integrity through two mechanisms as follows:-

(a) Controlling the sequence in which the application programs can be run (function (A) in the figure). In engineering systems, the sequence in which application programs are to be run influences the design of the database structure (see Sect 6.4.1). Therefore, once the database structure has been set up, it is imperative that the sequence initially planned is adhered to.

(b) Automatic recognition of the ramifications of running a specific application on other data in the database (function (C) in the figure). It has been shown in Sect 6.4.3 that the structure of a typical mechanical product displays a hierarchy. Due to the iterative nature of the product development process, the data relating to a particular subassembly may change frequently. In each case, the validity of much of the product-modelling data relating to its subassemblies and components will immediately become uncertain. It is therefore important that such affected data be automatically identified by the system. The logic for the identification of the affected data groups can be deduced from the sequence planned from the application programs including the iteration loops.

Once the suspect data groups have been identified, the following alternative courses of action become possible can be arranged:-

- remove such data from the database.
- where some of the application programs are of long duration, such data can be retained on the system as constituting an old version of the design and examined as and when the related application programs are re-run.
- where the number of levels in the hierarchyof the product structure is low, it may be feasible to test the validity of such data immediately.

Automatic modification of such data will rarely be possible in engineering systems because most of the application programs require input from the user, based on his expertise. In general, the application programs with which the affected data was created initially will need to be re-run.

The design of the control module requires an expert knowledge of the the realization methodology for the product. Thus, if either the desired sequence of the applications or the structure of the product is changed, the module has to be modified accordingly. The design and maintenance of the control module therefore constitutes an important additional function of the database adminstrator in a CAE environment.

6.5.3 **The CAE database structure**

Two important proposals are made regarding the structure of a CAE database. These are (a) a two-layered conceptual database structure and (b) a hierarchically structured 'global' database.

(a) **A two-layered conceptual database structure**

In the three-level architecture of the conventional database system presented in Sect 5.3.1 (see Fig 5.2), the conceptual view has been defined as a relatively permanent description of the data resource of an enterprise and an external view as a temporary one, derived from the conceptual view, to serve the requirements of a set of applications. Also, the general understanding is that only shared data i.e. data likely to be used by more than one application should reside in the database. In an integrated CAE environment however, certain factors prevail which necessitate a change to these rather purist notions. These factors are as follows:-

(1) Certain data aggregates exist which, though only ever used by one application, are best processed using database techniques. Catalogued data, such as materials data, are examples of these.

(2) A CAE database system needs to have the ability to

capture intermediate data and to support the iteration so prevalent in engineering applications (see Sect 6.4.2). Since the data used by a typical application program will span several files in the conceptual view, implementing these facilities with the conventional conceptual view would almost certainly jeopardize the integrity of the data in it.

(3) In a CAE system, the same representation of an object may be required by more than one application. Where this representation is different from that in the conceptual view, it is desirable that the database system has the ability to retain it separately so as to avoid the overhead of regenerating it from the conceptual view on every occasion that it is required by other applications.

These factors led to the proposition of a conceptual database configuration consisting of one higher-level 'global' database and several lower-level local databases as shown in Fig 6.11. It is emphasized that the proposed layering into a global and local databases is only with regard to the conceptual view. The local databases are therefore not equivalent to and do not replace the external views in the conventional three-level database architecture.

The design of the global database should be focussed on the requirements for the integration of the entire CAE

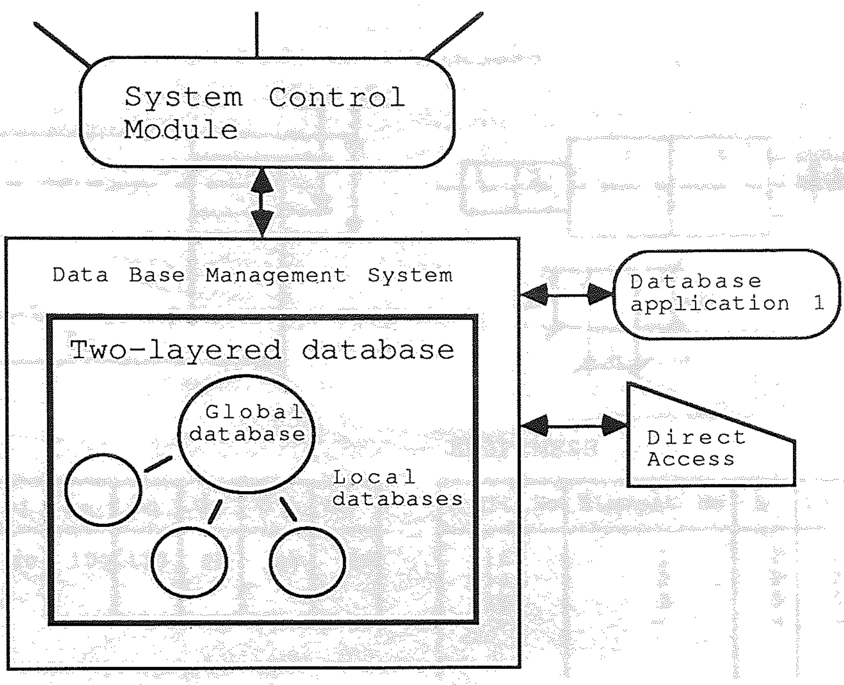
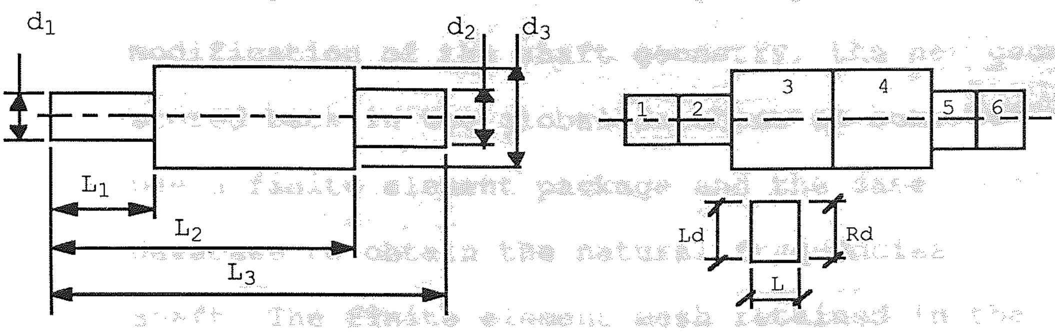


Fig 6.11 Depicting the two-layered conceptual database structure proposed for integrated CAE systems

environment. It should therefore contain the following types of data:-

- Truly common data i.e. data likely to be used by a number of applications which are otherwise not closely related.
- Concise representations of objects, in a form suitable for relating with other common data.

The requirements implied by the factors (1), (2) and (3) listed earlier in this section are fulfilled through the use of local databases. The functioning of a typical local database is described below using an example.



SHAFTGEOM

Shaft No	L_1	L_2	L_3	d_1	d_2	d_3
16	40	100	190	25	45	50

SHAFTMESH

Shaft No	Element No	L	Ld	Rd
16	1	20	25	25
16	2	20	25	25
16	3	30	50	50
-	-	-	-	-

(A) Shaft representation in global database (B) Representation of same shaft in a local database

Fig 6.12 Showing two different representations of the same object in a multi-layered schema

Consider a CAE system involving the design and manufacture of shafts. The design of the shaft may proceed as follows:-

- Use a drafting program and its associated local database to develop an initial shaft geometry. On completion, store the geometry data in a global database file such as SHAFTGEOM shown in Fig 6.12(a).
- Use a Finite Element package and its associated local database to analyze the stressing and deflection of the shaft. The required FE mesh is generated from the shaft

data in the global database and is stored in a local database file such as SHAFTMESH shown in Fig 6.12(b). Assuming that the stress analysis package permits the modification of the shaft geometry, the new geometry is stored back in the global database on completion.

- Use a finite element package and the same local database to obtain the natural frequencies for the shaft. The finite element mesh retained in the local database may be used thereby avoiding its regeneration from the data in the global database.

The iteration within and any storage of intermediate data required by any of the application programs cited is accomplished through the associated local databases. The temporary product-modelling data such as that in SHAFTMESH would be discarded only when the design of the shaft is considered to be completed.

(b) **A hierarchically structured global database**

The observation that the structure of a typical discrete mechanical product displays a hierarchy (see Sect 6.4.3) can be used to advantage in the design of the global database propounded in (a) above. Once the required product has been described in terms of a hierarchy, the following benefits can be realized:-

(1) A large number of entities of interest become obvious immediately i.e. the assembly,

subassemblies and individual components. The identification of entities of interest is one of the more difficult tasks in conventional database design.

(2) A large number of relationships of interest can be recognized immediately e.g. the 'is part of' relationship between a lower-level subassembly or component and its owner.

(3) A consistent method for identifying each and every subassembly and component is suggested by the hierarchy i.e. the identifier attribute set for a subassembly or component can be made up from that of its owner and the attributes that differentiate the owned entities. Thus in the example shown in Fig 6.7, the shafts and bearings can be identified using the schemes

(Gearbox No, Shaft assembly No)and
and (Gearbox No, Shaft assembly No, Bearing No).

The hierarchical description of the product will only assist the identification of the entities and relationships of interest regarding the structure of the product. Clearly, other data of interest will exist in the CAE system which cannot be recognized directly from the structure of the product. The following heuristic procedures are proposed for the identification of such other data groups of interest:-

(1) The categories of data that can be associated with a typical mechanical component are shown in Fig 6.13. Some of the data groups shown may also be

Data category	Examples
1. Spatial	- 2D models (projections, drawings) - 3D models (Surface, CSG) - Interobject data e.g. tolerances
2. Physical	- Material data e.g. classification into metal plastic., material physical & technological properties
3. Structural	- Standard part or assembly data - Assembly description i.e "goes into" information - Functional description - Product family data - Group technology data i.e. "is similar to" data.
4. Technological	- Manufacturing description e.g machining operations, tools, raw materials, NC programs
5. Commercial & business	- Customer(s) - Order quantities - Stock levels - Suppliers

Fig 6.13 Showing the categories of data associated a typical mechanical product

applicable to subassemblies in a product. Each leaf in the hierarchical description i.e. each component in the product can be examined with regard to each of the data groups shown.

(2) The data entities and relationships of interest initially identified can be be used in a data entity matrix as shown in Fig 6.14 to identify other relationships of interest. Each cell in the chart represents a potential relationship

between the row and column entity or relationship. Each of them can therefore be examined. Any useful relationships recognized is added to the

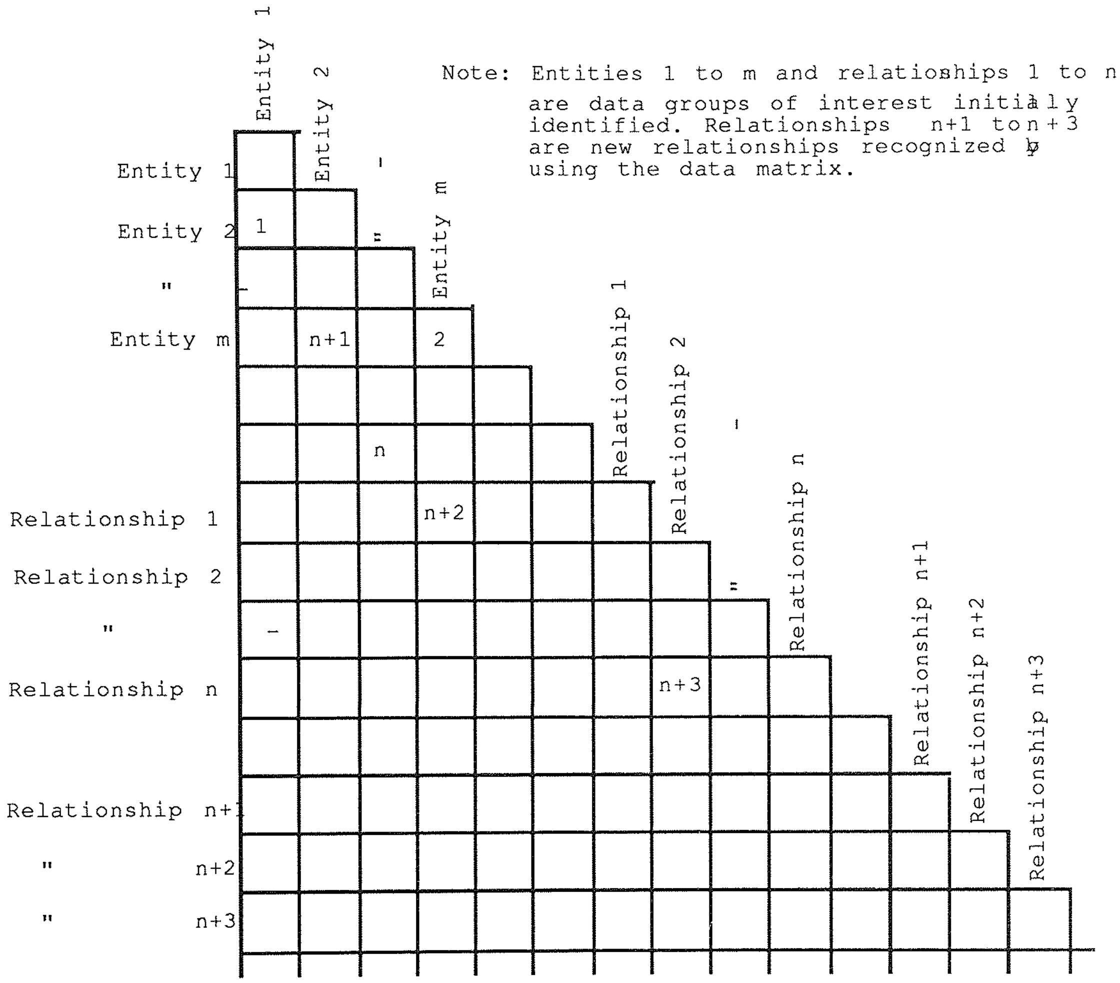


Fig 6.14 Showing a data matrix that can be used to identify new data relationships of interest in an environment

rows as well as to the columns so that it can be contrasted with all other existing entities and relationships.

In general, the procedures proposed above will lead to the identification of data entities to be modelled in the global database. The subsequent design of the local databases will be based mainly on the requirements of different groups of applications in the system.

Once the data entities and relationships of interest have been identified, the design of the database can be continued as described in Sect 5.4. As with the design of the CAE control module described earlier, an expert knowledge of the realization methodology for the product is a prerequisite for an optimum design of a database structure.

6.6 **Closing remarks**

The proposals made in this chapter regarding an integrated CAE database system have been based on three main premises i.e. (a) that basic conventional database tools can be used satisfactorily for the management of business and production engineering data, (b) that the current problem area is the processing of the more dynamic product-modelling data in engineering design activities and (c) that for purposes of integration, the requirement is for a database system in which both types of data can be managed efficiently. A study of the special characteristics of CAE data and engineering activities was therefore conducted, which lead to the incorporation of two new features in the conventional database system architecture i.e.the CAE control module and a two-layered conceptual database model.

No attempt has been made in the chapter to enumerate either the data entities or the applications that should constitute an integrated system as this would have implied the design of a specific system similar to those advertised in current trade literature. Instead, the focus has been on the development of a generalized database system architecture and a database design methodology that would allow the design and implementation of bespoke integrated systems. This approach seems to be just what is required for integration in manufacturing enterprises that have already invested in computer aided technology.

The proposals made would seem to provide partial solutions to the problems that currently hinder CAE integration as discussed in Chapter 3. These new concepts were incorporated in an experimental integrated CAE system developed as part of this research work. This is described in the next chapter and an evaluation of the proposals is then presented in Chapter 9 based on the performance of the experimental system.

C H A P T E R S E V E N

AN INTEGRATED CAE SYSTEM FOR THE DESIGN OF INDUSTRIAL GEARBOXES

7.1 Introduction

A generalized methodology for the realization of a discrete mechanical product was formulated in the first phase of the research work as outlined in Chapter 3. In the second phase, database concepts and technology were recognized as offering a viable approach to the management of information in integrated computer aided engineering systems based on the methodology. Their deeper study lead to the proposals made in Chapter 6 regarding an integrated CAE database system. The next requirement was therefore to evaluate the proposals made in the first two phases: the strategy adopted was to apply them to a specific computer aided engineering task. An experimental integrated CAE system designed and implemented for the purpose is described in this chapter.

The sample CAE task chosen was the design and manufacture of simple industrial gearboxes. There were two main reasons for the choice. Firstly, the general principles of operation as well as the design and manufacturing requirements and procedures for industrial gearboxes are likely to be easily understood, if not already known by most mechanical engineers. The design and usage of the package could therefore be focussed on its information management aspects without being clouded by the technical complexity of the product. The second reason was that two programs, one for spur and helical gear design and a suite for Finite Element analysis of bars, beams and torsional members, were already available in the department and could be adapted for gearbox design. This enabled a greater computing effort to be focussed on integration issues than on the developmemt of new application software. In the end, three other application programs i.e. for drafting of shafts, bearing selection and shaft material specification, and the CAE control module described in the previous chapter needed to be developed in order to create an integrated CAE package of adequate scope for the purposes of the research.

The description of the CAE system is arranged as follows. Firstly, a tailor-made methodology for the design and manufacture of gearboxes, developed from the generalized one given in Chapter 3, is presented. The overall structure of the implemented system is then outlined, followed by more detailed descriptions of its main operational components i.e. (a) the application programs, (b) the control module, (c) the database schema and (d) the data directory/dictionary system. An evaluation of the system is presented in Chapter 8.

7.2 A methodology for the design and manufacture of simple industrial gearboxes

The primary function of a gearbox in a machine is to change the torque-speed ratio or mechanical advantage between the output shaft of the driving unit (typically a prime mover) and the input shaft of the driven unit. The term 'industrial gearboxes' is used here to denote the range of gearboxes used in the drive mechanisms of industrial machinery such as pumps, conveyor belt drives etc., excluding those used in instrumentation equipment and vehicles which tend to be very specialized.

7.2.1 The configuration of an industrial gearbox

Fig 7.1 shows a plan section of one of a number of shaft assemblies in a typical industrial gearbox. Brief notes are given below regarding each of the main parts shown.

(a) **Gears**

Many different types can be used, the most common being spur, helical, worm and bevel. The design of gears is commonly based on standards such as BS 436 - 1967, AGMA 218.01 (1982). The gear design program used in the research is based on the former.

Gears are normally designed in-house and then either manufactured or selected from gear catalogues.

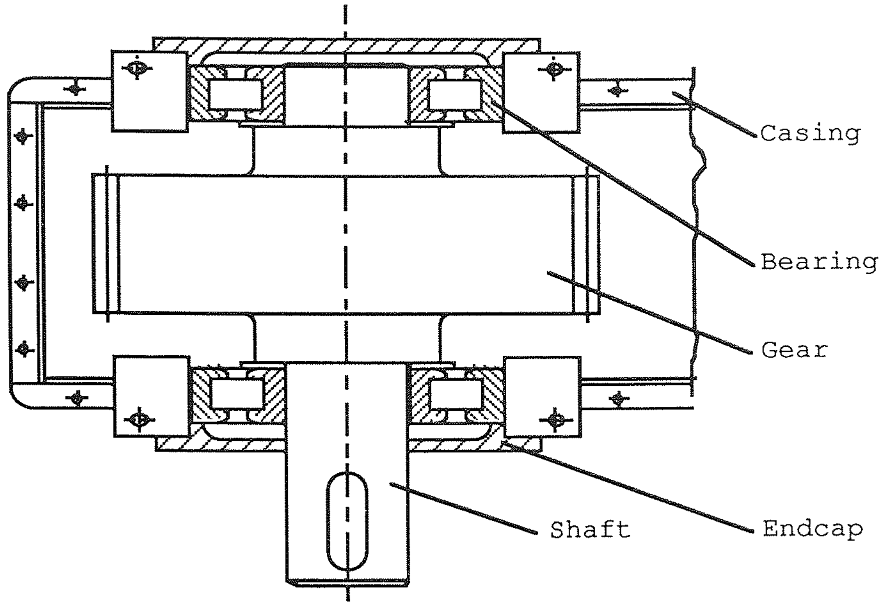


Fig 7.1 Showing a plan section of one of several shaft assemblies in a typical industrial gearbox

(b) **Shafts**

Shafts are used in a great variety of shapes depending on the nature of the loads applied through the the gears and the conditions at the bearing supports. The design of shafts is a basic engineering design problem involving considerations of static and fatigue strength, stiffness and deflections including the prevention of whirling. The shaft analysis programs used in this research are based on the Finite Element method.

Both the design and the manufacture of shafts can generally be accomplished within the enterprise.

(c) **Bearings**

In general, only rolling contact ball or roller bearings are used in industrial gearboxes.

Self-aligning bearings are commonly selected because of their ability to withstand the misalignment that can occur due to bad assembly after maintenance or overhaul. Other factors considered in the selection of the bearing type include the the magnitude and direction of loads, rigidity, available space and limiting speeds while the subsequent selection of bearing size is generally based on its load carrying capacity and the requirements regarding life and reliability.

The bearing selection program created as part of the research work is based on the bearing life equation given by the SKF Bearings Group for selection from their product range.

(d) **Casings**

Apart from the ability to support the shaft assemblies, an important consideration in the design of casings is the lubrication of the gears and, in particular, the dissipation of the heat generated. Casing design is not included in the CAE package created.

Other components not shown in Fig 7.1 may include keys to locate the gears on the shafts, bearing seals to exclude dirt and retain the lubricant and the bolts and nuts to secure the gearbox to a base.

7.2.2 Synthesis of a methodology for gearbox design and manufacture

In the course of developing a product, the activities shown in Fig 3.1 cannot, may not necessarily all be carried out within the same enterprise. Commonly, some design and production work is contracted out and some of the more specialized components, such as bearings, have to be bought from other manufacturers as standards. The range of activities undertaken internally will therefore vary from one enterprise to another depending on many factors such as the area of specialization of the company, the availability of capital for investment in hardware and software and calculations of economies of scale.

For the purposes of this research work, an hypothetical enterprise was envisaged whose main activities with regard to gearbox manufacture are limited to the following:-

- Gear design and specification. The gears are subsequently purchased from a specialist manufacturer.
- Shaft design and machining.
- Bearing selection.
- Design of the casings. The manufacture of the casings is contracted out.
- Assembly of the gearboxes. Standard bolts, nuts and gaskets and seals are used.

In addition to these predominantly CAD/CAM activities, production and business functions such as planning and marketing respectively, are also carried out as normal.

Fig 7.2 shows an adaptation of the general product realization methodology depicted in Fig 3.1 to a specific one for the design and manufacture of gearboxes in the hypothetical enterprise. It should be noted that because many components are duplicated in different locations in a gearbox assembly, some of the activities shown, such as shaft design, and the iteration loops associated with them may need to be carried out more than once.

As discussed in Chapter 4, it is generally not yet possible to computerize all the activities shown in Fig 7.2. Notwithstanding this however, in the implementation of a database system in the hypothetical enterprise, an important objective would be to structure the system such that even those activities which are not currently computer-aided can make use of the stored data, perhaps through a query facility, and that if and when it becomes possible to computerize more activities, minimum changes to the system would be needed. This approach is reflected in the design of the overall system as presented in Sect 7.3 and of the control module and database structure as described in Sects 7.5 and 7.6 respectively.

In a practical context, the INPUT-PROCESS-STORAGE-OUTPUT charts shown in Figs 3.3 to 3.14 would then be used at this

STEPS IN THE GENERAL METHODOLOGY FOR DISCRETE-PRODUCT REALIZATION (See Sect 3.5.4, also Fig 3.1)	METHODOLOGY FOR GEARBOX DESIGN
1. Recognition of a need	Not relevant. The need for torque-speed variation in power transmission sytems is well recognized in engineering
2. Product selection	Not relevant. The use of gearing for torque-speed variation in power transmission systems is well established in mechanical engineering

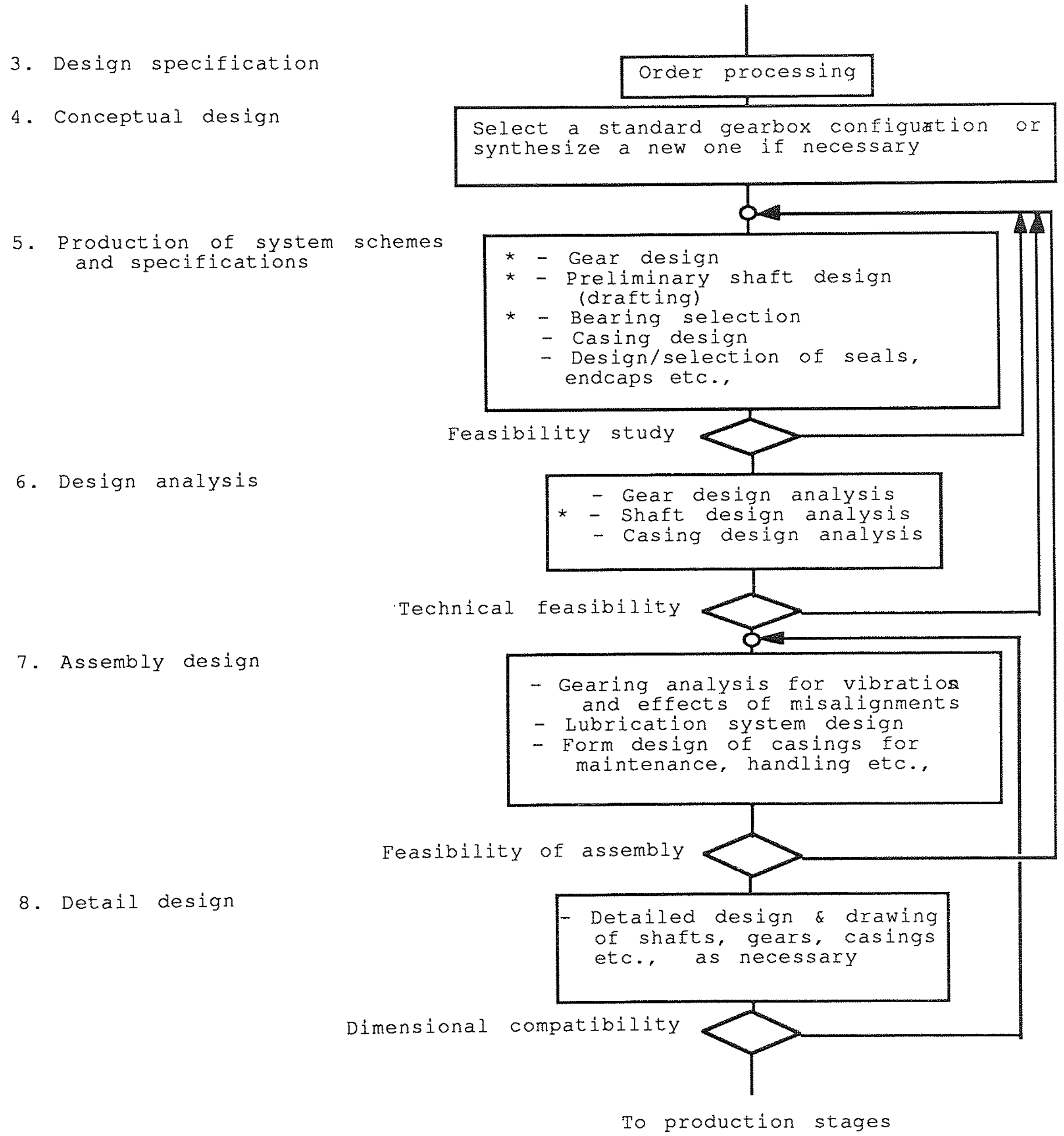


Fig 7.2 Depicting a methodology for the design of industrial geaboxes

stage to analyse the data transformation requirements of the product-specific methodology. In this research work however, such a full analysis would have been too speculative and perhaps unnecessary as well. The analysis was therefore limited to the requirements of those activities for which appropriate software was either already available or could be developed within the scope of the research undertaken. These are marked with asterisks in Fig 7.2.; the data associated with them is discussed in conjunction with the design of the conceptual database (Sect 7.6). The description of the CAE control module given in Sect. 7.5 (see Fig 7.4) and that of the database structure presented in Sect 7.6 (see Fig 7.11) both indicate how they can be extended to incorporate other applications. More specific guidelines for expanding the scope of the package are also outlined in Sect 8.3.

The design and implementation of a CAE database system to support the parts of the design and manufacturing process indicated is presented in the rest of the chapter.

7.3 **Configuration of the integrated CAE database system**

The overall structure of the integrated CAE database system created is shown in Fig 7.3. The concept used is essentially that proposed in Chapter 6 except that both the control module and the data directory/dictionary system have been implemented using the DBMS's own programming language.

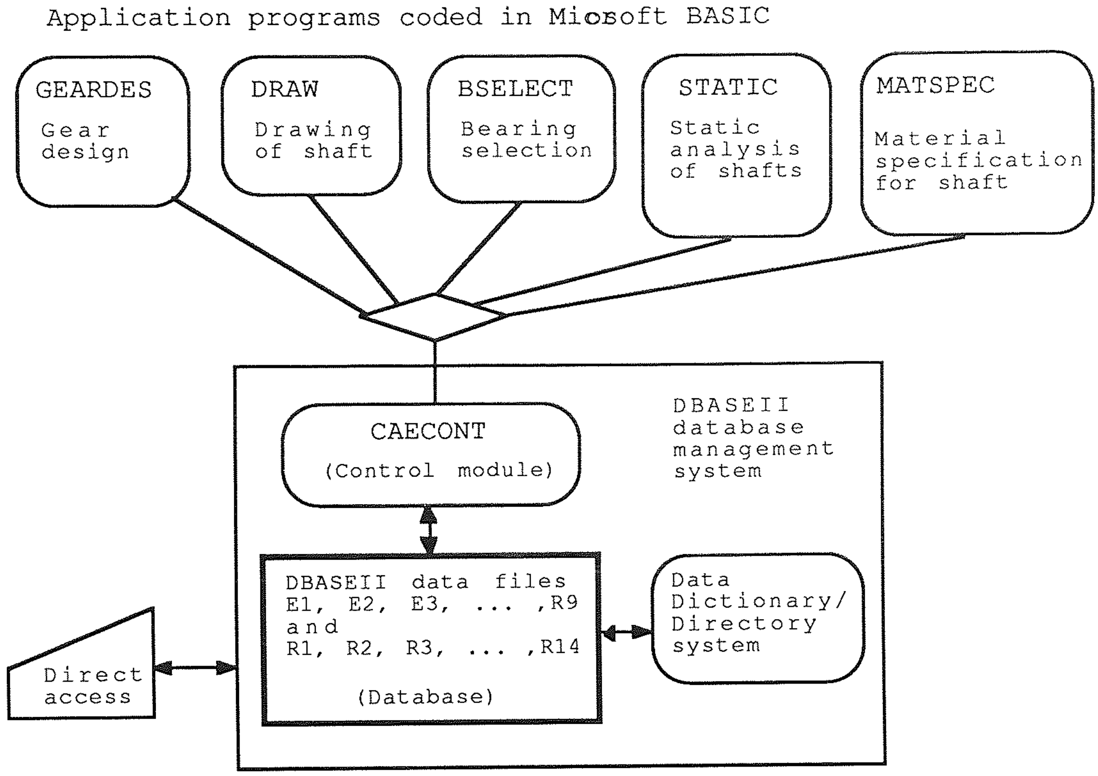


Fig 7.3 Showing the configuration of the integrated CAE database system implemented

7.3.1 **The computer and database management systems**

The package has been implemented on an Apricot Xi personal computer (256K RAM, 10Mb Winchester disc, MSDOS 2.11 operating system including GSX graphics system extension, Microsoft BASIC interpreter).

A proprietory relational database management system 'dBASE II' (MSDOS vers 2.4) was chosen for the package for two reasons. Firstly, it was considered at the time to be the most widely used relational DBMS on personal computers and, secondly, a version compatible with the computer system used was readily available on the market. An extended discussion of dBASE II is given in Appendix 1. Its main specifications are as follows:-

Records per file (i.e. tuples)	65535 max
Characters per record	1000 max
Fields per record	32 max
Characters per field	254 max

dBASE II is based on relational algebra and has its own programming language. During the period of the research work, an enhanced product 'dBASE III' came onto the market but, by then, the implementation work for the CAE system had advanced beyond the stage where it would have been advantageous to switch over to the new package.

7.3.2 **Programming languages and interfacing**

All the application programs shown in Fig 7.3 are coded in Microsoft BASIC and and the MSDOS graphics extension system (GSX) is used extensively to effect graphics in some of the programs. On the other hand, the control module and the data directory/dictionary system are implemented in DBASEII's own programming language.

The interfacing between dBASE II and each of the application programs is not simple. Firstly, dBASE II commands cannot be embedded in high level language programs. Also, although a facility is provided in DBASEII to run a high level language program in a similar manner to a normal subroutine, it could given in Appendix 1. Its main specifications are as follows:-

Records per file (i.e. tuples)	65535 max
Characters per record	1000 max
Fields per record	32 max
Characters per field	254 max

dBASE II is based on relational algebra and has its own programming language. During the period of the research work, an enhanced product 'dBASE III' came onto the market but, by then, the implementation work for the CAE system had advanced beyond the stage where it would have been advantageous to switch over to the new package.

7.3.2 **Programming languages and interfacing**

All the application programs shown in Fig 7.3 are coded in Microsoft BASIC and and the MSDOS graphics extension system (GSX) is used extensively to effect graphics in some of the programs. On the other hand, the control module and the data directory/dictionary system are implemented in DBASEII's own programming language.

The interfacing between dBASE II and each of the application programs is not simple. Firstly, dBASE II commands cannot be embedded in high level language programs. Also, although a facility is provided in DBASEII to run a high level language program in a similar manner to a normal subroutine, it could not be used in this case because of inadequate main memory on the Apricot computer system (256K). Instead, the interfacing was achieved through an MSDOS batch program and a set of flag files written to disc. The transfer of data to and from the application programs is through the use of MSDOS system data files which can be created and read by both dBASE II and MSBASIC (as sequential files).

The other operational modules shown in Fig 7.3 are discussed individually in the subsections that follow.

7.4 The application programs

Five application programs are incorporated in the package as shown in Fig 7.3. While two of these were already available in the department on a different computer system, the three others were developed as part of the research work. The objective was to create a sufficiently complete CAE system that, in addition to serving the purposes of the research work , could be used subsequently as a teaching aid in design courses in the department.

There is no chaining or direct interchange of data among the programs: each is independently interfaced to the database through the control module as shown. Some introductory notes about each of the application programs are given in the subsections that follow. The programs are each discussed in greater detail in Appendix 2. As it would be inappropriate to include program listings in this thesis, program sizes are quoted in this section in order to indicate the amount of meticulous programming and editing effort involved in incorporating each of the programs into the integrated package.

7.4.1 **The gear design program GEARDES** (Size: 33K MSBASIC)

This is a program for the design of spur or helical gears to BS 436-1967. The main design criteria for such gears are the surface strength of the tooth and its bending strength at the root. These are discussed in more detail in Sect (a) of Appendix 2 where a summary flowchart for the program is also shown.

A version of this program was available on a BBC model 'B' computer in the department. Its incorporation into the CAE database system involved the following tasks:-

- Transfer from the BBC to the Apricot computer system. A direct transfer of the program as a text file was eventually possible after familiarization with the asynchronous communication software on both machines and the preparation of the appropriate cables.
- Recoding in Microsoft BASIC. The editing required a lot of time and care, particularly because BBC BASIC allows the use of subprograms (procedures) while MSBASIC does not.
- Inclusion of routines for transacting with the

database. This needed firstly, the analysis of the program's data requirements i.e. meanings of data elements and the formats of the data files, followed by the design of the subschema files after matching these to the contents of the CAE schema.

7.4.2 **The drafting program DRAW1** (Size: 24K MSBASIC)

This is a drafting program developed as part of the research work for the specific purpose of enabling the creation of the geometry of a completely axisymmetric shaft. The program enables easy creation and modification of the geometric data i.e. the length and end diameters for each element of the shaft and the fillet radii at appropriate nodes.

As MSBASIC does not provide direct graphics commands, the development of this program required a knowledge of the usage of the graphics capability provided by MSDOS operating system through its graphics extension system (GSX). A flowchart and a sample screen display are given in Sect (b) of Appendix 2.

7.4.3 **The bearing selection program BSELECT** (size: 48K MSBASIC and 20K dBASE II)

This program, which was also developed as part of the research work, uses the loading and geometry data (created by GEARDES and DRAW1 respectively) stored on the database to select appropriate rolling bearings for the shaft. It is based on the guidelines given by SKF (Catalogue 3000 III E/GB 666 III. June 1980) for selection from their range of products. The selection procedure is outlined in Sect (c) of Appendix 2. The flowchart for the program is also given in the same section.

The bearing data is stored in a number of dBASE II files which constitute part of a local database. Accordingly, while the selection parameters are prepared in the main program coded in MSBASIC, the actual selection function is effected through a DBASEII program. This transaction is limited to the local database only; it corresponds to the 'on-line' access facility shown in Fig 6.10. The files in the global database, such as the ones containing the geometry data, are updated only after the bearing selections and their effects on the geometry of the shaft have been accepted by the designer.

Although this particular program is based on SKF guidelines, the same main inputs parameters i.e. shaft loads, bearing locations the shaft speed and the required bearing life would be required for the selection of products from other manufacturers. It is therefore possible to extend the program to the selection of other manufacturers' products provided the corresponding bearing data is added to the database.

7.4.4 **The shaft analysis program STATIC** (Size: 52K total MSBASIC)

This is a chained set of programs which perform static analyses of the shaft taken as a bar, a beam and a torsion member. The programs are based on the Finite Element method for stress analysis. This method is briefly outlined in Sect (d) of Appendix 2. A flowchart for the program is also given in the same section.

The program set was created from three separate programs originally developed in the department for the static and dynamic analysis of bars, beams and torsional members. Its creation and inclusion in the CAE package involved the following work modules:-

- Transfer of programs from a BBC to the Apricot computer system and recoding in Microsoft BASIC.
  The comments made in Sect 7.4.1 apply.
- Inclusion of routines for transacting with the database. The data input section of the bar analysis program was modified to enable the retrieval of all geometric and loading data for the shaft. The other two programs were then modified such that they utilized the same input data initially retrieved. The output sections of the three

programs were also combined so as to display the axial, bending and torsional analysis results and store these to the database in a coordinated manner.

- Modification of the beam analysis programs to enable the analysis of bending in two planes.
  Since radial loads and moments acting on transmission shafting are in oblique planes, they need to be resolved in two perpendicular planes for purposes of bending analysis. The original beam analysis program was therefore altered to enable analysis in the vertical and horizontal planes.
- Development of a program to obtain the theoretical stress concentration factors 'K_t' for direct, bending and torsional stresses at a shoulder on the shaft. For this purpose, the stress concentration curves provided by **Peterson** [**58**] were digitized and a curve-fitting program was then used to approximate these with polynomial functions which were then used in the program.

7.4.5 **The material specification program MATSPEC**

(Size: 29K MSBASIC)

This program uses the stresses, stress concentration factors and the various fatigue strength reduction factors at the critical node on the shaft to give a specification of the appropriate material for the shaft. The program is based on the use of the Soderberg rule as discussed by **Peterson** [**58**] for the computation of a factor of safety for a machine element in combined tension, torsion and bending. The program enables the substitution for all known values in the equation, resulting in an expression of the safety factor as a function of the yield and ultimate tensile strengths only. This can then be used in conjunction with a materials catalogue to select the appropriate material for a required safety factor.

This program was also developed in the course of the research work. It is presented in greater detail in Sect (e) of Appendix 2.

7.5 **The CAE control module**

The main functions of the CAE control module have been outlined in Sect 6.5 (see Fig 6.10). The flowchart for the module implemented is shown in Fig 7.4. The main routines in the module are CAEMENU1, CAEMENU2 and the application-specific ones CAEGEAR, CAEDRAW, CAEBRGS,CAESHAFT and CAEMSEL. The module is coded in DBASEII's own programming language.

When using the CAE package, the user is aided by a series of self-explanatory menus produced by the control module. These show both the progress of the design as well as the courses of action open at any point in time. In this sense therefore, the control module makes the CAE package very user-friendly,

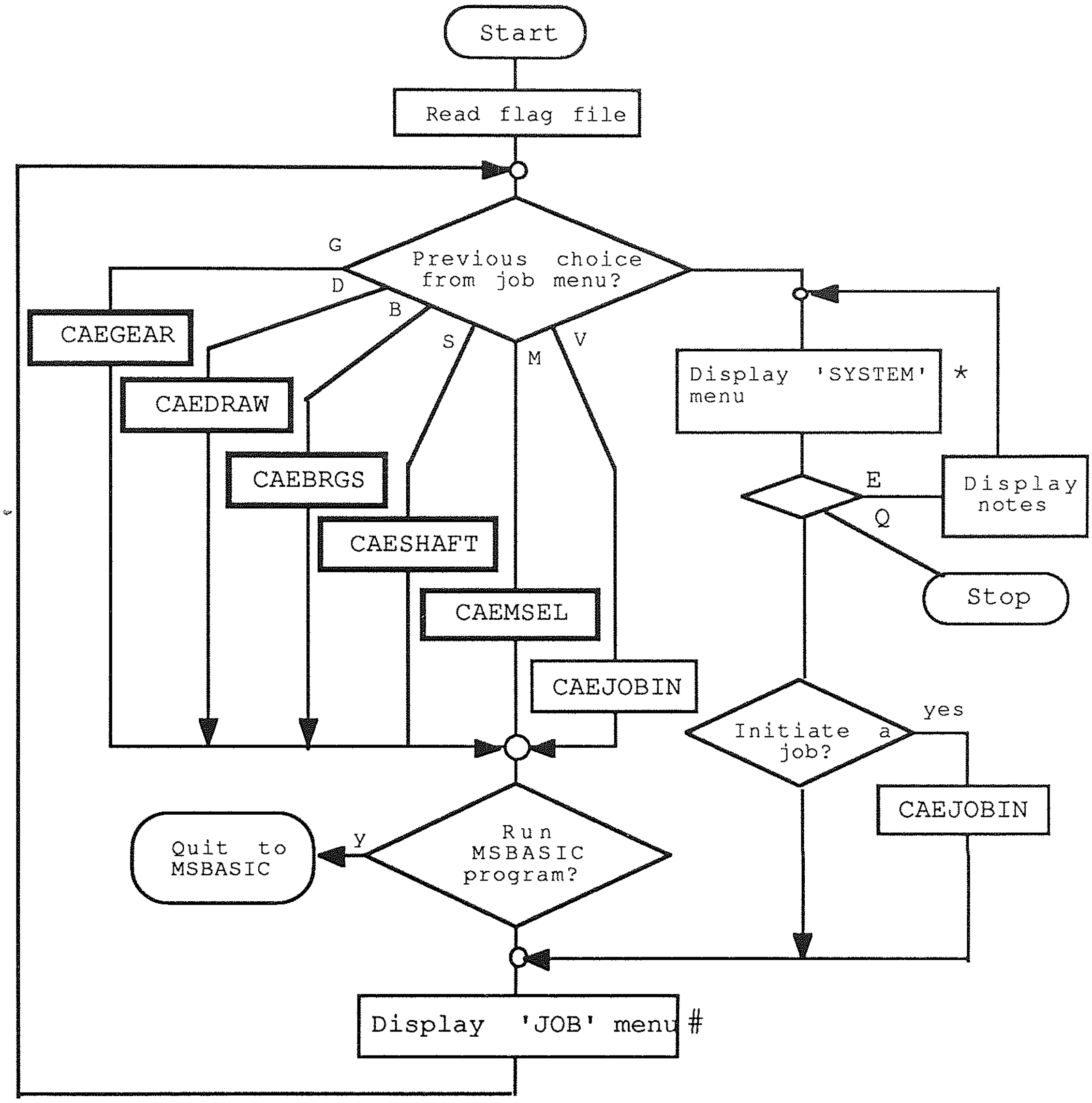


Fig 7.4 Showing the main flowchart for the CAE system control module

* <u>System Menu</u>

Explanatory notes
Continue with an existing job
Initiate or cancel a job
Quit integrated CAE system
(Choice 'E', 'C', 'I' or 'Q')

\# <u>Job Menu</u>

Gear pair design
Draw shaft
Bearing selection
Shaft analysis
Material specification
(Choice = 'G','D','B','S','M'
OR 'V' to view configuratio
OR 'Q' to quit this job)

eliminating the need for a user manual.

The module has been designed to enable the user to operate at three levels i.e. at the 'system', 'job' or 'function' level, thus reflecting the typical scenario in a manufacturing environment as depicted in Fig 6.9. The operation of the module is briefly described below in terms of these levels.

(1) **The system level**

When the CAE system is first started, the user is presented with the system menu as shown in Fig 7.4. The user can then choose to:-

- Read explanatory notes about the CAE system. These are displayed on the screen after which the system menu is presented to him again.
- Initiate or cancel a job. Initiation of new job involves the generation of a job identifier e.g. EMP7 (meaning Job No.7 under the code EMP) and the selection of a standard gearbox for the job. The job menu (see (2) below) is then presented to the user.
- Quit the system i.e. leave the gearbox design system.
- Continue with an existing job. A routine to identify the required job is run after which the job menu (see (2) below) is presented.

(2) **The job level**

Once the job required has been identified (or one has been initiated), the job menu shown in Fig 7.4 is presented. The subprogram CAEMENU2 also displays the present state of the job with regard to each of the five functions shown on the menu e.g. whether the analysis of the shafts has been completed, is incomplete or has not been started at all. The options open to the user are also worked out and displayed. The user can then choose to:-

- Perform any of the five functions shown e.g. design gears, draw shaft etc., depending on which options are open (see (3) below).
- View the details of the standard configuration for the particular job he is working on. The MSBASIC program ('CONFIGS') which displays the standard configurations is run, after which the job menu is redisplayed. This is done through the subprogram CAEJOBIN shown in the figure.
- Quit working on the particular job in which case the system menu, providing the user with the options listed under (1) above.

(3) **The function menu**

If the user chooses to perform one of the functions shown on the job menu, the related interface subprogram CAEGEAR, CAEDRAW, CAEBRGS, CAESHAFT or CAEMSEL is run. These subprograms all have the same logical structure as that shown in Fig 7.7 for CAEGEAR. Firstly, the details of the job configuration and other job data on the database are used to display the status of the current state of the job with regard to the particular function.

For example, if the user wishes to carry out the gear design function on a job which has two gear pairs, the program will display whether any of the gear pairs have already been designed. It also works out the options open to the user. He can then choose either one of the options or quit the particular function in which case the job menu is presented, providing him with the options listed under (2) above.

If the user wishes to proceed with the function, the data required by the particular application program i.e. the subschema data, is then retrieved from the database and the related application program is run. If the user chooses to store the results obtained from the application program, the database is updated as soon as the control module is re-entered. In all cases, the function menu is presented again, allowing the user to either continue with the same function e.g. design a second gear pair, or go back to the job menu.

The two main functions of the control module namely the sequencing of application programs and the recognition of the effects on the existing data of running a particular program, are performed mainly through the subprograms CAEMENU2 and each of the interface programs CAEGEAR, CAESHAFT etc., The operations of these programs are therefore presented in more detail in the subsections 7.5.1 and 7.5.2 that follow.

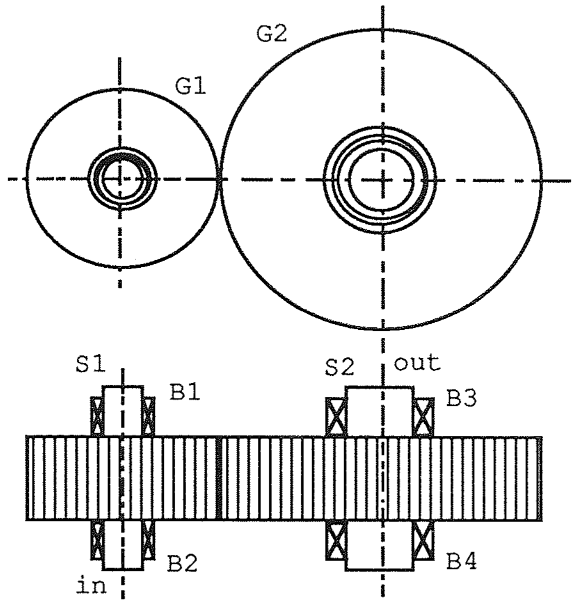
7.5.1 The job menu subprogram CAEMENU2

In the CAE database system developed, each design job is based on one of two standard gearbox configurations as shown in Fig 7.5. This limitation is discussed in Chapter 8. The subprogram CAEMENU2 uses the configuration data and the job progress data kept on the database to display the status of the the job and also work out what options, in terms of applications, are open to the user at any time. A general flowchart for the subprogram and a sample job menu are shown in Fig 7.6.

7.5.2 The interface subprograms CAEGEAR, CAEDRAW, CAEBRGS,CAESHAFT and CAEMSEL

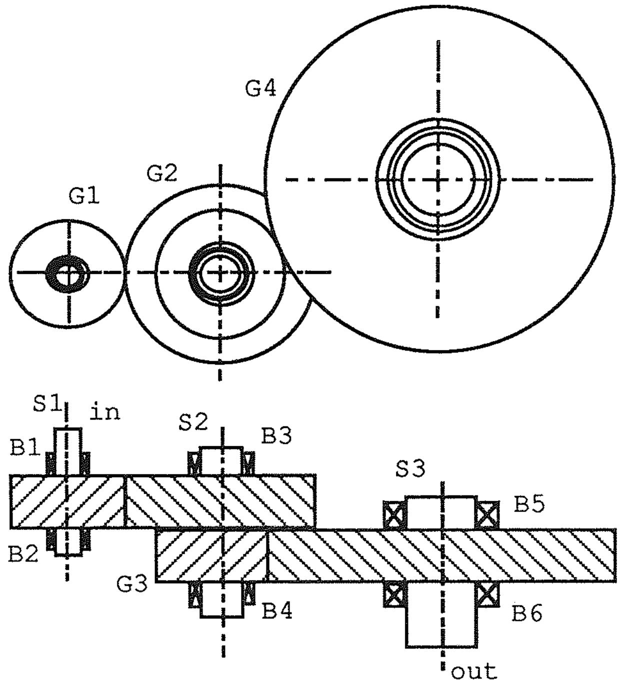
Each of these subprograms shown in Fig 7.4 interfaces an MSBASIC application program to the database structure. Since they all have the same logical structure, their operation is described here in terms of only one of them namely CAEGEAR which is the interface for the gear design program GEARDES.

The general flowchart for the subprogram is shown in Fig 7.7. The subprogram is run either after the gear design option has been chosen from the job menu or immediately after dBASE II is re-entered after running the gear design program GEARDES. In the latter case, the database is first updated with the new gear design results and the job progress data is also



CONFIGURATION No.1

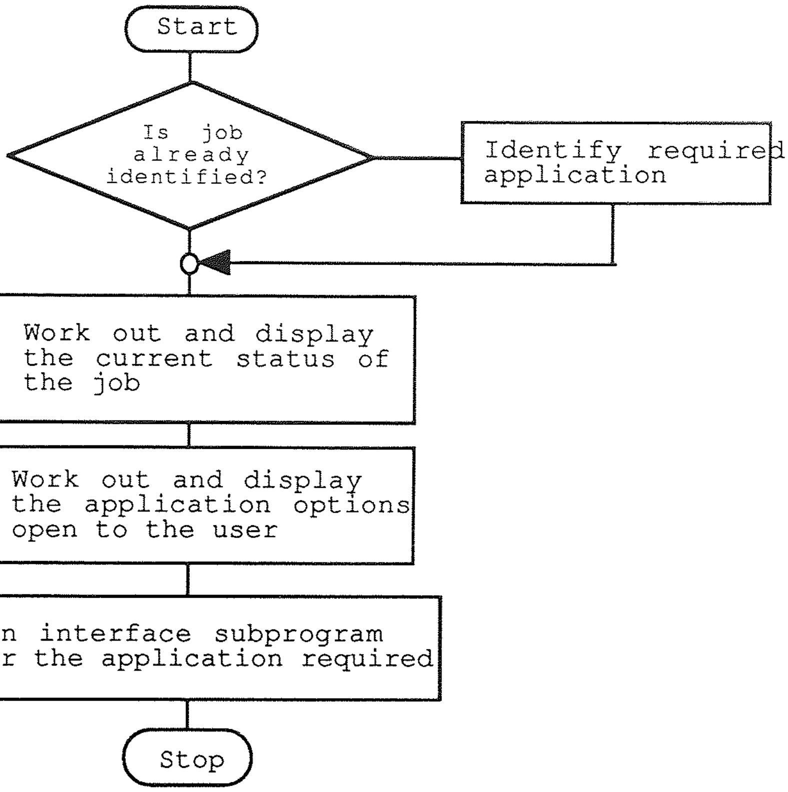
Single reduction
2 spur gears (G1 & G2)
2 parallel shafts (S1 & S2)
4 self-aligning ball
bearings (B1 to B4)



CONFIGURATION No.2

Double reduction
4 helical gears (G1 to G4)
3 parallel shafts (S1 to S3)
6 taper roller bearings
(B1 to B6)

Fig 7.5 Showing the two standard gearbox configurations modelled in the CAE database system



Integrated Computer Aided Engineering System

INDUSTRIAL GEARBOX DESIGN

Job level

Job Number EMP1 Configuration: No.2

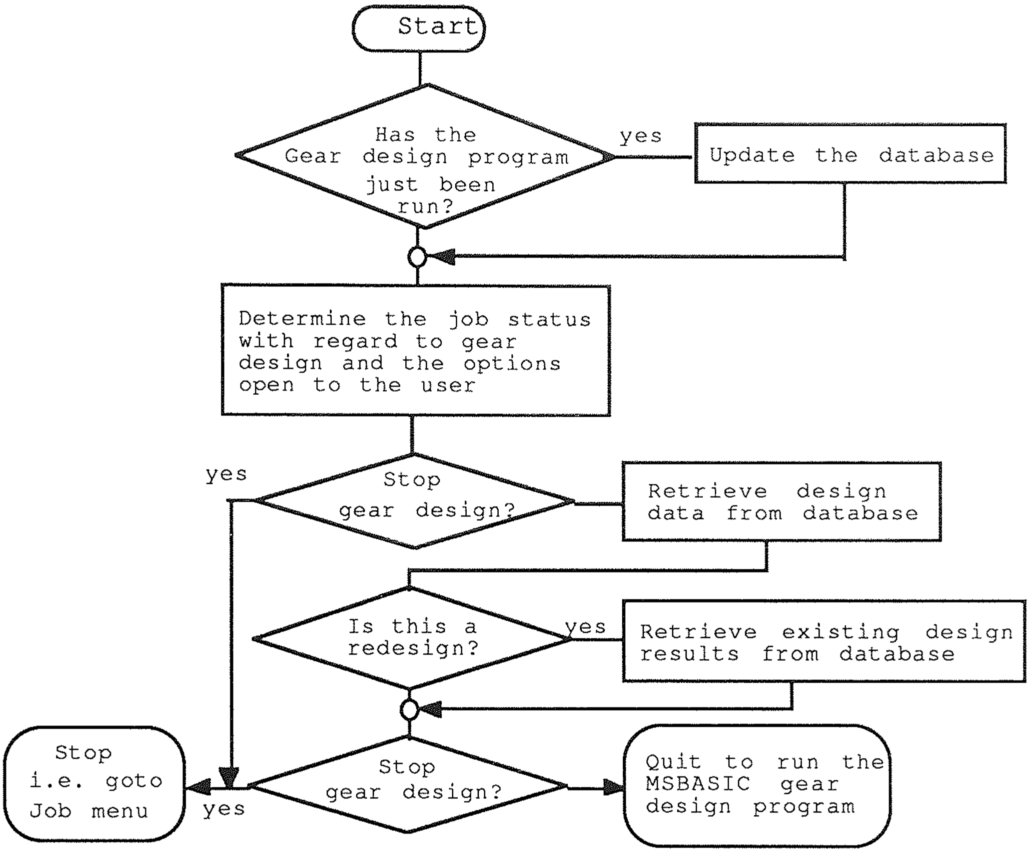
Date started: 10/07/86

Job module	State
Gear pair design	Completed
Draw shaft	Incomplete
Bearing Selection	
Shaft analysis	
Material specification	

Enter the required option (G,D,B)

OR enter V=view configuration, Q=quit this job

Fig 7.6 Showing the general flowchart for the subprogram CAEMENU2 and a sample job menu



Integrated Computer Aided Engineering System

INDUSTRIAL GEARBOX DESIGN

Application level

Job Number EMP1 Configuration: No.2
Application: GEAR DESIGN Date started: 10/07/86

Function module	State
Design gear pair No.1	Completed
Design gear pair No.2	Completed

Enter the required option (1,2)
OR enter 'Q' to discontinue with this application

Fig 7.7 Showing the general flowchart for the interface program CAEGEAR and a sample application menu

modified. In the case of a redesign, all existing design results are replaced by the new data. However, although other affected data on the database can be identified by using the configuration data, a facility for erasing, versioning or immediately validating such data as proposed in Sect 6.5.2 has not been included in the package. Instead, this redudant data is left on the database until it is replaced when the application programs that generated them in the first place are run again. This arrangement was purely for expediency reasons i.e. inclusion of such a facility was considered to be of little immediate interest in the research work: it is proposed later as a possible improvement to the package.

Fig 7.7. also shows a sample function menu presented after both status of the job and the user options in relation to the gear design function have been determined. In the case of a job based on configuration No.2, the function menu will show whether one or both gear pairs have been designed. Also, with this configuration, the design of gear pair No.2 cannot preceed that of gear pair No.1 since the gear design program requires a pinion speed as an input parameter. Such expert knowledge has been built into the subprogram that works out the options open to the user.

7.6 **The CAE database**

The CAE system implemented is based on a relational database. On a physical level, the database consists of a set of dBASE

II data files, each holding data about an entity or relationship required in the process of designing a gearbox. On a conceptual level, each of these files represents a relation in third normal form as discussed in Chapter 5. The details of the contents of these files are specific to the design of gearboxes and are therefore of marginal interest for research purposes. Of greater importance is the general applicability of the procedure and techniques deployed in the development process for the database. These are therefore outlined in this section. Some details of the database contents are given in Appendix 3.

The design followed the general procedure proposed in Sect 5.4 which has four main stages as follows:-

(a) Requirements formulation and analysis
In the experimental CAE system, this was not an important step since the main functions required to be supported by the database structure were well defined i.e. design of gears, drawing of shafts, selection of bearings, static analysis of shafts and specification of shaft materials. Also, the main groups of data associated with these functions could be easily identified i.e. the loading, geometries and standard materials for gears and shafts as well as bearings data. In an industrial environment however, this would be a crucial stage since the boundaries of the database system would not be so well defined in advance. The

proposals made in Chapter 3 regarding functional and data analysis would then need to be followed in a more deliberate manner than was necessary in this particular application.

(b) Conceptual design

The Entity-Relationship approach briefly introduced in Sect 5.4 was adopted at this stage. The process and the results obtained are summarized in Sect 7.6.1.

(c) Implementation design

The normalized relations generated during conceptual design were directly implemented as dBASE II data files. The main data description and data manipulation commands used are listed in Appendix 1 in conjunction with a brief general description of the database management system itself.

(d) Physical design

This was of marginal concern in this research: the use of a proprietory DBMS and a computer system with a large disc storage capacity (10Mb hard disc) gave a satifactory system performance with little need to optimize, for example, the record formats or data manipulation routines.

7.6.1 **Conceptual database design**

The Entity-Relationship approach followed is depicted in Fig 7.8. Each of the steps is described below.

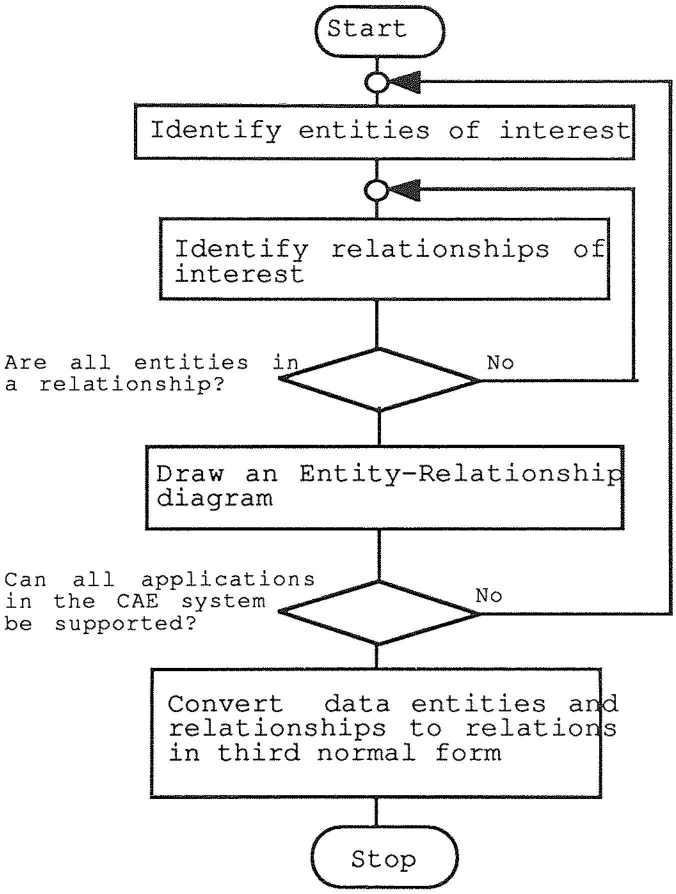


Fig 7.8 Depicting the Entity-Relationship approach used in the design of the conceptual database

(a) Identification of entities of interest.

A set of entities were identified from the hierarchical description of a typical industrial gearbox as shown in Fig 6.7 and the use of the checklist given in Fig 6.13. These entities are listed in Fig 7.9. Note that only those entities needed to support the applications

Entity name	Identifier	Entity Function
E1*	Job number	To hold data about a specific job, such as date of initiation and the configuration. Also holds job progress data.
E2	Configuration number	To hold data describing each of the standard gearbox configurations used in the CAE system
E3	Gear material name	To hold various properties of standard materials for gears
E4*	Gear number	To hold data about each of the gears in a gearbox.
E5	Catalogue gear number	To hold data about standard gears from a specialist manufacturer i.e. a gear catalogue
E6	Shaft material name	To hold various properties of standard materials for shafting.
E7*	Shaft number	To hold data about each of the shafts in a gearbox.
E8	Bearing number	To hold data about each of the bearings in a gearbox
E9	Catalogue bearing designation	To hold data about standard bearings from a specialist manufacturer i.e. a bearings catalogue

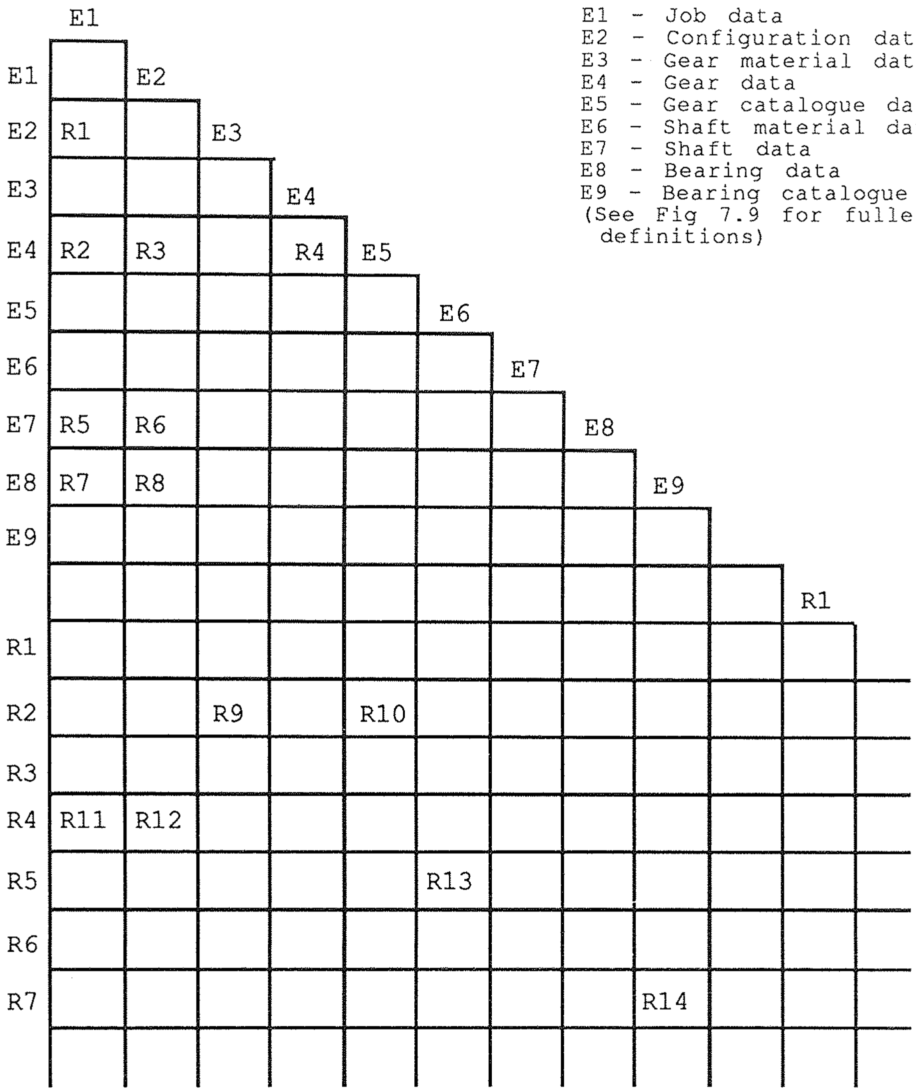
Fig 7.9 Showing the entities of interest identified during the design of the conceptual database

(Identified from the hierarchical description of the gearbox)

actually implemented in the system were considered. Others could be included as and when the user applications become available.

(b) Identification of relationships of interest.

The entities identified were used in a data matrix to identify relationships of interest. The matrix is shown in Fig 7.10.



E1 - Job data
E2 - Configuration data
E3 - Gear material data
E4 - Gear data
E5 - Gear catalogue data
E6 - Shaft material data
E7 - Shaft data
E8 - Bearing data
E9 - Bearing catalogue data
(See Fig 7.9 for fuller definitions)

Fig 7.10 Showing a data matrix used to identify relationships of interest during conceptual database design

The meanings of the relationships can be deduced from the figure. For example, R4 is a relationship between a gear and another gear i.e. a gear pair. It is required in the conceptual dabase in order to model data such as centre distances and gear ratios, which are attributes of a gear pair rather than of an individual gear. Again, only relationships of immediate interest were considered.

(c) Generating the Entity-Relation diagram.

The E-R diagram developed is shown in Fig 7.11. Such a diagram gives a very concise description of the current status of any database and is therefore a very useful tool for the maintenance and subsequent expansion of the database.

(d) Development of normalized relations.

This process had two important stages. Firstly, the key and non-key attributes sets were determined for each entity and relationship shown in the E-R diagram. This was followed by the normalization of the resulting relations to third normal form. The relation schemes developed can be examined by using the data directory /dictionary system developed (see Appendix 4).

The layering into a global and a number of local databases as proposed in Chapter 6 was not significant in the simple system developed. This was primarily because all the applications implemented are very closely-related. This is discussed further in Chapter 8.

The design of the conceptual database was followed by that of the external views pertaining to each of the applications. These are very specific to each of the programs and are therefore of little interest in this thesis. Their design is therefore not discussed here.

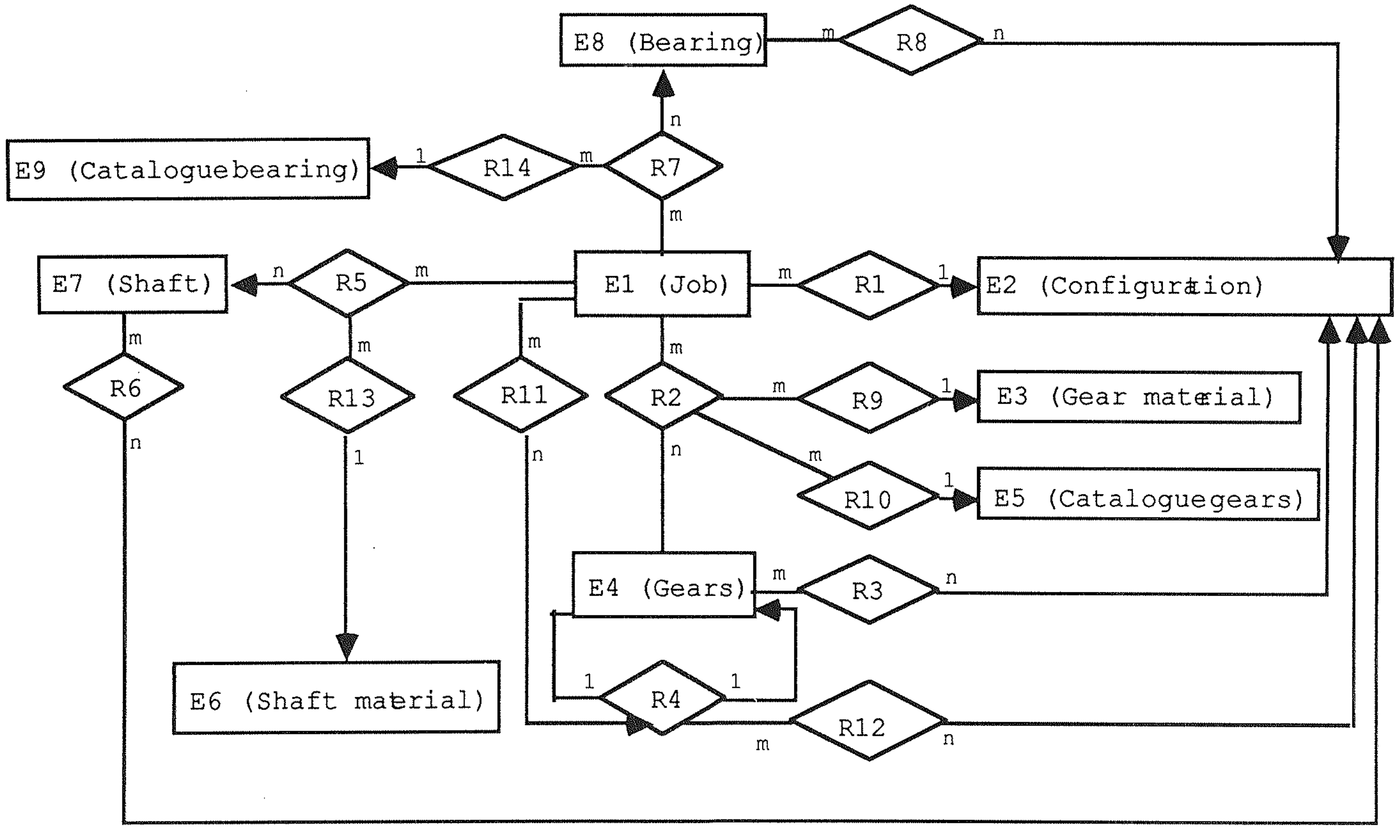


Fig 7.11 Showing the Entity-Relationship diagram developed during during the design of the conceptual database

7.7 **The data dictionary/directory system**

A D/DD system was developed in the early stages of the implementation work for the CAE system, to serve two important functions as follows:-

- enable the examination of the contents of the database at at any time.
- exercise control over modifications to existing data structures as well as over the creation of new ones. this facility is necessary in any database system if the integrity of the data is to be maintained.

The design of the system and its implementation into a working suite of programs is outlined in Appendix 4.

7.8 **Miscellaneous remarks**

The implementation of the CAE database system required a more substantial and meticulous effort than what might be implied by the design concept shown in Fig 7.3. The following tasks required a considerable amount of programming work.

(a) Translation of the programs initially based on the BBC computer system into MSBASIC on the Apricot system, and the design of new programs as detailed in Sect 7.4.

(b) The design of the CAE control module and its implementation using the dBASE II programming language. In particular, the implementation of the subprograms for storing and retrieving data (called by the interface programs) required thorough acquaintance with the DBMS's data manipulation commands.

(d) The design of the data dictionary/directory system and its implementation using dBASE II programming language.

The development of software per se, was not an objective of the research work: the main concern was with the optimum configuration of the database system and the design of the database structure so as to facilitate integration in CAE systems. The programming tasks involved are enumerated above in order to indicate the type of work that may need to be done in the process of establishing a database system in a real engineering environment.

The CAE database system described in this chapter was based on the proposals made in Chapter 6. Two gearbox design examples are presented in Appendix 3 to indicate the usage and capabilities of the whole integrated CAE system and also show the type of output given by each of the five application programs. An evaluation of the package is given in the next chapter, leading to the general conclusions of the thesis presented in Chapter 9.

C H A P T E R   E I G H T

EVALUATION OF THE INTEGRATED COMPUTER AIDED ENGINEERING PACKAGE

8.1 Introduction

The CAE database system for gearbox design described in the previous chapter was created as a vehicle for experimenting with the proposals made in Chapter 6 regarding integrated systems. The design of the package was based on general specifications derived from a study of the characteristics of engineering data and activities. The experience with it can therefore be used to evaluate the general approach proposed in this thesis regarding the processing of information in integrated CAE systems i.e. the use of database concepts and technology: this evaluation is presented in Chapter 9. The objective in this chapter is to assess the package created with reference to the general requirements in an integrated CAE system as discussed in Chapter 6.

The chapter is arranged as follows. Firstly, the designation of the package as an 'integrated' system is justified, followed by a brief discussion of its scope. The compliance of the system with the design specifications formulated in Chapter 6 is then assessed, after which the general applicability of the procedures used in its design and implementation is examined. The value of the package as a research tool and the possible improvements to it are then discussed in the last part of the chapter.

8.2 Designation of the package as an 'integrated' CAE system

In Chapter 4, an integrated computer aided engineering system has been defined as one in which a set of computer tools are linked together, the objective being to optimize their collective operation in a bid to boost the productivity of the whole engineering system. In the package, five applications i.e. gear design, drafting, bearing selection, finite element stress analysis and shaft material specification are linked together through the use of a common database.

The applications implemented in the experimental CAE system cut across different areas of expertise and may therefore conceivably be carried out in different sections and perhaps on different computer systems in the design department of a large manufacturing enterprise. All the applications in the package happen to belong to just one area of activity i.e. the design field. Also, whilst all the programs developed and used are quite sophisticated, it is not difficult to imagine far more complex packages in use. Nevertheless, each of them can exist as a separate tool. In addition to providing many of the general advantages of the database approach as discussed in Sect 5.2.1, the linkage of the five applications through the database system i.e. their integration, was seen to improve the efficiency of the gear design process in the following specific ways:-

- The main design parameters are entered only once by the designer. When these are required by subsequent applications, they are retrieved from the database instead. This feature not only minimizes the manual data input effort and therefore the errors that can be associated with it, but also ensures the consistency of the common data throughout the design. As an example, the figures for the power rating and shaft speed entered during the design of the very first gear pair in a gearbox are automatically retrieved for the design of the other gear pairs. Also, the resulting shaft loads are retrieved from the database for the bearing selection and shaft analysis applications.

- The deductive features and the design progress information provided by the CAE control module virtually eliminate the manual documentation that might otherwise be required in order to keep track of the design of a particular gearbox. This enables the user to concentrate on the more technical aspects of the design and therefore can only enhance the quality of the end product.

8.3 The scope of the gearbox design system

As discussed in Sect 6.1, the main objective in creating the package for gearbox design was to enable an objective evaluation of the ideas developed in the research regarding the integration CAE systems. The eventual scope of the package was therefore determined, in the main, by its ability to satisfy this overriding requirement.

The scope of the package created can be assessed in terms of three important considerations, namely (a) the generality of the package as a gearbox design system, (b) the range of design applications implemented and (c) its conditions of use.

(a) **The generality of the system**

This refers to the range of gearboxes that can be designed using the system.

The package enables the design of gearboxes based on the two standard configurations shown in Fig 7.5. also, the application programs incorporated in it restrict the design to the use of spur and helical gears, completely axisymetric shafts and self-aligning ball, spherical roller or taper roller bearings. Despite these present limitations however, the concepts underlying the design of the package are sufficiently general to allow its enhancement to incorporate other configurations and other types of machine elements.

The range of gearbox configurations can be extended in two different ways. The easier way would be for the database adminstrator to adopt a number of other gearbox configurations as standards and describe these to the database system by adding records to the entity 'E2' and the associated job-independent relationships 'R8', 'R6', 'R13' and 'R12' (see Fig 7.11). Alternatively, an application program can be developed to enable individual designers to generate job-specific configurations and have these automatically described to the system in a similar manner.

More types of gears and bearings as well as more generalized shaft geometries can be accommodated by adding records and new attributes as necessary to the entities 'E3' to 'E9' and to the relationships associated with each of them (see Fig 7.11).

(b) **The range of design applications implemented**

The wide range of tasks required for the complete design of a gearbox are indicated in Fig 7.2. Those implemented in the package are marked with asterisks in the figure. The other applications that may need to be incorporated in order to create a more elaborate systems include the design of casings and endcaps, the selection of seals, and the design of the lubrication system.

The implementation of such a wide range of applications was considered to be beyond the scope of this research although more would have been included in the package had the appropriate programs been readily available. As in (a) above, the CAE database structure has been designed in such a way that more applications can be included in the package if required. Apart from the modification of the CAE control module, the interfacing of a new program will, in most cases, only require the addition of records and attributes to the existing data entities and relationships: these will not affect the running of the existing programs. In a few cases however, new entities and relationships of interest may need to be introduced into the conceptual view. The techniques proposed in Sect 6.5.2 can be used to great advantage in such cases.

(c) **The conditions of use**

The experimental CAE system has been implemented on a single-user and single-tasking computer system. It however seems possible to apply exactly the same concepts to the design of larger industrial systems perhaps requiring the application of distributed database technology. A viable configuration would be to install the global database and control module on a master computer while maintaining the local databases and related application programs on a set of departmental ones.

A number of the computers in an industrial system may provide multi-user and multi-tasking facilities. Unlike the production and business activity areas where parallel processing is generally possible, such facilities are unlikely to offer many extra benefits in design activities such as the ones included in the package. This is due to the procedural relationships among design data groups as discussed in Sect 6.4.1.

The brief discussion presented in this section has shown that it is to talk in finite terms about the scope of the CAE system created since its design concept enables it to be expanded with relative ease when so required. The overall observation is that the differences between the experimental system and one envisaged in industry are more to do with scale than with structure.

8.4 Compliance with the design specifications

The analysis of the nature of engineering data, engineering activities and discrete mechanical products carried out in this research (see Sect 6.4.1 to 6.4.3) led to the formulation of a number of important design specifications for integrated CAE database systems as enumerated in Sect 6.4.4. These are in two categories: general ones relating to the whole system and those specific to the database structure. Following is an evaluation of the compliance of the implemented CAE system with these specifications.

8.4.1 General system specifications

Five requirements were recognized as being very important for purposes of integration. These are examined below.

(a) **Retention of conventional database manipulation facilities**

This has been ensured by the use of a contemporary data model i.e. the relational model, and a reputable commercially available database management system (dBASE II). The introduction of a control module and a two-layered database in the conventional database system architecture is for purposes of managing product-modelling data only: they do not impose any restrictions on the manipulation of other types of data that may be stored on the database.

(b) **Support for a variety of object representations**

No instance of multiple object representation has been implemented in the package. The proposition of a two-layered database structure arose from a realization that in CAE systems, there may arise a frequent need to support several versions of the same object on the database. No such need arose in the case of the gearbox design system created because of the relatively small number of data entities modelled in the global database and the simple manner in which this data is used by the application programs interfaced to the database.

The closect case of multiple object representation that could have been implemented is the maintenance of two versions of the geometry of a shaft as in the example discussed in Sect 6.5.3 (see Fig 6.12). Although the Finite Element version is required by both the bearing selection program 'BSELECT' (for the calculation of bearing loads) and the static analysis program 'STATIC', its generation from the global version is relatively simple and is therefore best carried out by routines within each user program. However, if the geometry of the shaft had been more complicated (e.g. non-axisymetric or integral with a gear) as to require the running of a separate mesh generation program for example, or needed to be modified repeatedly by a number of user programs, it would then have been necessary to maintain the Finite element version explicitly in a local database. Such requirements are likely to arise in more elaborate systems.

(c) **Consistency maintenance across object representations**

There is no case of multiple object representation implemented. This requirement is therefore inapplicable in this case. In principle though, it would have been possible to maintain such consistency by creating tailor-made DBMS subprograms to map between the global and local versions of an object. These would be activated only when a local version of the object had been accepted by the designer.

(d) **Distributed data processing facilities**

No distributed processing facilities are provided on the system. Although the Apricot Xi computer system on which the package is implemented has facilities for asynchronous communication with other systems through the RS232 port, the scope of the package did not raise a need for distributed processing.

(e) **High-level language interfaces**

The database management system used (dBASE II) provides facilities for interfacing with high-level languages such as BASIC and FORTRAN. In the package, all the application programs are coded in Microsoft BASIC. Main memory size permitting, dBASE II provides a facility to run high-level language programs in the form of subprograms. For example, the expression

QUIT TO 'MSBASIC GEARDES','DBASE GEARSAVE'

would cause the BASIC program GEARDES to be run, followed by that of the dBASE II program GEARSAVE. In the package however, this facility has not been used because of inadequate main memory. The interfacing with the BASIC programs has been achieved through a batch file. The main penalty suffered because of this was the extra programming effort required in creating the batch file and setting up of a flag file on disc to enable the execution of the control program to resume at the right point after the running of an application program. As far as the overall operation of the package is concerned, the method of interfacing devised does not seem to have resulted in any major drawbacks.

The interchange of data between dBASE II and the MSBASIC programs has been achieved through the following dBASE II commands:-

COPY TO Temp.**txt sdf delimited**

and **APPEND FROM** Temp.**txt sdf delimited**

The first expression copies the contents of a previously opened dBASE II file into a system data file called 'Temp'. This can then be read as a sequential file in MSBASIC. The second expression appends the contents of a sequential file called 'Temp' into a previously opened dBASE II file. Considering the generic pattern of engineering activities as discussed in Sect 6.4.2 (see Fig 6.5), the interfacing effected in the package would seem to be quite satisfactory for many practical purposes.

8.4.2 **Data structure requirements**

Two specific requirements were identified regarding the structure of the database as follows:-

(a) **Flexibility**

This refers to the ability to extend or modify the structure with minimum disruption of the system. The database implemented is based on the relational data model. Because of its implicit expression of relationships among groups of data as described in Sect 5.4.3, the relational model generally offers levels of flexibility which are difficult to achieve with other models.

The design of the database is such that it can be extended to accommodate other data entities and relationships of interest. For example, suppose a need arose to implement an order processing application in the package. An obvious entity of interest is the order itself, perhaps identified by an order number and with data such as order date, customer name etc., as the non-prime attributes. This can be integrated into the database structure by, for example, creating a relationship between it and the job data entity ('E1' in Fig 7.11), with the identifier attribute set being made up of an order number and a job number. After generating the non-prime attributes for the new entity and relationship and normalizing as necessary, the resulting relations can be implemented as dBASE II data files and interfaced to the order processing program. Note that the existing data files and programs will not be affected at all by such an extension of the database structure.

Apart from the creation of new relations, extra attributes can be added to the existing ones without needing to modify the user application programs. For example, if the gr design program was to be extended to enable the design of the hub, new attributes modelling the hub dimensions could be added to the existing gear data entity 'E4' in Fig 7.11 without affecting the operation of the programs that currently use this entity.

In contrast to the relative ease with which the database structure can be extended, its modification by, for example, the removal of relations or attributes or the alteration of their names, would involve a great deal of editing work. This is because the relation and attribute names are used as variables in the dBASE II routines that manipulate the data in the database. These subprograms would therefore require modification as well. This was found to be particularly disadvantageous during the design of the system when a database file structure needed to be modified frequently in a bid to optimize the operation of the whole system.

(b) **Hierarchical structure**

The hierarchical description of the gearbox configurations used has only two levels i.e. the gearbox assembly as the top and the component types as the bottom. This is demonstrated in the Entity-Relationship diagram (Fig 7.11) where the bearing data (E8), the shaft data (E7) and the gear data (E4) are shown directly related to the gearbox data (E1) rather than to a 'shaft assembly' as suggested in Fig 6.7. This simple description was found to be adequate for the gear design system: a more refined description could have been used if necessary.

8.5 Compatibility with the nature of engineering data and activities

In addition to providing the opportunity to test the ideas developed in the research regarding CAE system integration, the creation of the gearbox design package also enabled an examination of the validity of the characteristics accorded to engineering data and activities earlier in the research work. These are reviewed briefly in this section in relation to the data and programs in the CAE system created.

8.5.1 **Properties of engineering data**

Five important characteristics had been recognized as follows:-

(a) **Complex data structure**

The Entity-Relationship diagram (Fig 7.11) shows fourteen relationships of interest among the nine data entities identified. Furthermore, much of the product-modelling data kept on the database are attributes of these relationships rather than the entities. For example, much of the gear design data is modelled in the relationships 'R2' and 'R4' rather than in the gear data entity 'E4' (see Appendix 3).

Additional complexity of the data arises from the fact that many of the relationships of interest are of the many-to-many type: this makes the normalization of engineering data that much more difficult.

(b) **Recursive definition of data objects**

Because of the limited scope of the package, no case of recursive data object definition arose in the database design process. In principle, it should be possible to cope with this property by extending the idea of a hierarchical description of mechanical engineering products propounded in this thesis.

(c) **Procedural relationships among data entities**

In the process of designing a gearbox, the design of the gears has to preceed that of the shafts on which the gears are mounted. Also, a complete design of the shaft requires its loading data. In the database implemented, there is therefore a procedural relationship between the data groups 'E4' and 'R4' which model the gear design data, and the data entity 'E7' which holds the shaft data. In the package, these relationships are maintained by the sequencing of applications imposed by the control module.

(d) **Versioning**

No facilities for versioning have been implemented in the package (see also Sect 7.5.2). Such facilities were considered to be outside the main interest of the research although they would be very desirable enhancements of the CAE system. the design of the database is such that these can be implemented in the future.

(e) **Complex data types**

dBASE II offers very limited data typing facilities: these are detailed in Appendix 1. Although these facilities were adequate for the requirements in the experimental system, more data types would be needed in more elaborate CAE systems. For example, it might be advantageous to store the stiffness matrix of a structure as a single unit of data rather than as a set of tuples in a relational database.

8.5.2 **Characteristics of engineering activities**

Six significant properties have been cited in Sect 6.4.2 as follows:-

(a) **Storage and retrieval of large units of data**

In the package, the interfacing between the application programs and the database has been designed such that all the data required by a particular program is moved from the schema to the subschema files before the program is called by the control module. A similar arrangement applies to the storage of data once the program has been run successfully. This arrangement proved to be very satisfactory. Even if the DBMS had offered the more efficient host-language interfacing, an arrangement for piecemeal storage and retrieval of data would have not been preferable.

(b) **Coupling of programs to the database**

Out of the five application programs incorporated in the package, only one i.e. the bearing selection program 'BSELECT' requires a closer coupling to the database beyond that required for data storage and retrieval as discussed in (a) above. This program requires repetitive access to the catalogue bearing data kept on the database. This seems to confirm the observation made in Sect 6.5.2 that, except for access to job-independent data for selection purposes, engineering design applications need not be as closely coupled to the database system as is required for commercial applications.

(c) **The ability to capture intermediate data**

The application programs in the package are all of short duration. With such programs, the need to store intermediate data i.e. data being generated in the course of running a particular program, is minimal since they can be rerun in a relatively short time if required. No such facility has therefore been implemented.

(d) **Validation of data during transaction with the database**

No facilities have been implemented for checking the validity of the data retrieved from or stored on the database apart from those provided within each of the application programs. The onus has been left to the expert user to ensure that the input data and design results presented to him at the start and end of each of the programs respectively are correct. The implementation of such facilities requires an expert systems approach involving a thorough definition of the validity limits for different types of data modelled in the database.

(e) **The need for a comprehensive query facility**

There is no comprehensive facility provided for handling queries involving data values: the data directory/ dictionary system implemented only deals with metadata such as the function and structure of a database file as well as those of the fields in it.

Direct access to the schema is possible via data manipulation commands provided by dBASE II: this however requires exact knowledge of where the data of interest is stored in the database. A more efficient way to view product-modelling data is to run the program with which it was created in the first place: in the case of a redesign, all the programs in the package display the existing design data. A comprehensive query facility could be of benefit in more elaborate systems.

8.6 General applicability of the methodology followed in the implementation of the integrated CAE system

During the design and implementation of the package, an important requirement was that both the procedure and the individual techniques used were applicable to the design of database systems for the realization of other discrete mechanical products. The fulfilment of this requirement is assessed in this section.

The creation of the CAE database system for gearbox design can briefly be described to have involved four main phases i.e. (a) development of a specific methodology for gearbox design, (b) the analysis of data for database design, (c) the design and implementation of the CAE control module and (d) the design and implementation of the database structure itself. This procedure would seem to be generally applicable. The generality of the techniques used in the various phases is discussed below.

(a) **Development of a realization methodology for a specific product.**

The general methodology formulated in Chapter 3 (see Fig 3.1) was used to develop a specific one for the design of industrial gearboxes as shown in Fig 7.2. This could have been done for any other mechanical product, the level of difficulty depending on the complexity of the product.

(b) **Analysis of data for database design**

The approach developed in this research regarding the design of integrated CAE systems requires an expert knowledge of the methods and therefore the main data groups of interest in the product realization modules to be supported by the envisaged system. Therefore, as in this particular application, a formal analysis of the data along the lines proposed in Chapter 3 will not be necessary in every situation. In every case however, the guidelines given can be used to analyze the data of interest in a more systematic manner.

(c) **The design and implementation of the CAE control module**

The general structure of the control module as shown in Fig 7.4 would seem to be appropriate for most discrete mechanical products. In particular, its operation at three levels i.e. 'system', 'job' and 'application' levels seems to enable a very smooth progression through the design of the gearbox.

(d) **The design and implementation of the database structure**

The design procedure as outlined in Sect 7.6 would seem to be generally applicable. So do the various techniques used in the design process to identify the entities and relationships of interest i.e. the hierarchical description of the product, the use of the checklist given in Fig 6.13 and the data matrix shown in Fig 6.14.

It is emphasized that the generality discussed in this section concerns the methodology and techniques used in the design of the database system and not necessarily the database structure developed. It is the belief of the author that in striving for integration in today's CAE systems, the major requirement is for effective system design methodologies as opposed to specific data structures.

8.7 **The value of the package as a research tool**

The discussion so far in this chapter has concentrated on the quality of the package as an integrated CAE database system. The creation of the package was however primarily meant to serve the purposes of the research work. Its value as a research tool is therefore assessed briefly in this section.

As stated in Sect 8.1, the main reason for creating the package was to allow experimentation with the proposals made in the research regarding integrated CAE systems. Its success in this aspect can be judged at two levels as follows:-

- Broadly speaking, the creation of an integrated package that gives an acceptable level of performance is a demonstration of the validity of the analysis carried out in the research work with regard to the feasibility of using contemporary database tools to effect integration in CAE systems. More importantly, the flexibility of the package created shows the generality of the ideas that guided its design and therefore their applicability to the more elaborate systems likely to be required in industry.

- The design and implementation of a specific CAE package revealed the relative significance of the characteristics of engineering data and activities analyzed earlier in the research. This helped in the determination of an order of importance among the specifications formulated for the design of integrated CAE systems. For example, the need to maintain procedural relationships among groups of data in a CAE database (satisfied through the CAE control module) was shown to be of greater importance than the provision of facilities for versioning or processing of ad hoc information queries. This relative importance would have been difficult to determine without experience with an actual system.

Two more general benefits provided by the implementation exercise can be cited. Firstly, the substantial data analysis and programming work involved helped to create a more realistic perception of just what the implementation of a CAE database system in an industrial environment would entail. It would not have been possible to gain this insight from a conceptual view of the proposed system as shown in Fig 6.8(b).

The second benefit afforded was to gain a good insight into the current state of database technology. The dBASE II database management system was chosen for the research primarily because it was the most well established relational system available on microcomputers at the time: its performance was therefore indicative of what could be expected of other systems. A discussion of dBASE II is presented in Appendix 1.

The discussion presented in this section leads to the overall conclusion that the implementation of a specific package was essential for the research work. It not only provided the opportunity to both demonstrate and evaluate many of the proposals made concerning the design of integrated CAE systems, but also indicated an order of importance among them and the realities that would need to be faced in implementation of an integrated CAE system.

8.8 **Possible improvements to the package**

The package as it stands served the purposes of the research work. It however has the potential to be developed further, not only by extending it to include some aspects of computer aided manufacture but also by expanding its scope in the design area itself. Such expansion would enable, if not necessitate, the implementation of more of the proposals made in Chapter 6 such as versioning and the layering of the database.

The scope of the package as a design system can be extended by enabling the use of more gearbox configuration and machine elements as proposed in Sect 8.2. Apart from expanding the scope, an important enhancement required is the incorporation of a facility to precede the design of a new gearbox by a search of the database for an existing design of similar specification. Even if such a design, if found, could not be adopted for the new job, the designer could use it as a check or guide in the design of the new gearbox.

The generality of the concepts on which the design of the package was based, have given it the flexibility to be enhanced in many different ways. The package therefore constitutes a very solid experimental 'rig' for continued research in integrated computer aided engineering systems.

CHAPTER NINE

GENERAL DISCUSSION AND CONCLUSIONS

9.1 Introduction

The main goal pursued in this research was the integration of applications in a computer aided engineering environment. The major developments in computer technology over the last decade and the increased general awareness of its potential have greatly promoted the notion of integrating all mainstream activities in an engineering enterprise. For pragmatic reasons, the research work was focussed on the requirements for the integration of design applications for discrete mechanical products. Nevertheless, the concepts developed were sufficiently general to allow future incorporation of the downstream manufacturing applications as well as many of the supporting activities in the total product realization process.

After a critical study of the process from the recognition of a need, through design and manufacture, to the marketing for a mechanical product, the main requirements identified for integration was the shareability of data among the various activities involved in the process. This led to the proposition that the database concept, so far developed primarily for the processing of commercial data, could be extended to the engineering design field in order to provide the data shareability required for integration. A subsequent detailed study of the particular properties of engineering data and activities revealed the need to introduce two new features in the architecture of a conventional database system i.e. a control module and layering of the conceptual database as discussed in Chapter 6. These new concepts were then embodied in an integrated CAE database system for the design of industrial gearboxes, in order to enable their objective evaluation. The main conclusions arising from the research work outlined above are presented in this chapter.

The conclusions reached are in two main categories (a) general ones relating to the strategy for integration and (b) those specifically about the database approach proposed as a route to integration. These are presented in Sect 9.2 and 9.3 respectively. Proposals regarding the direction of future academic research in integrated CAE systems are then presented in Sect 9.4 to mark the end of the thesis.

9.2 **The general strategy for integration**

The objectives and potential benefits of integration as well as the many problems that hinder its implementation in today's computer aided engineering systems have been discussed at length in Chapter 4. In this research, a survey of the literature and the experience with the CAE package developed in the course of the research work led to the identification of two major requirements for integration. They are (a) an adequate analysis of the functions required to be integrated and the data associated with them and (b) the mapping out of an implementation strategy for the integrated system. These are discussed separately in the subsections that follow. Both emphasize the critical role of the user enterprise in creating the framework for integration. The main idea that emerged was that it is up to the enterprise to provide the data structure and powerful DBMS facilities required to maximize the convenience with which the various stand-alone packages such as FE or CAPP can be linked into the system.

9.2.1 **Adequate functional and data analysis**

The design of a mechanical product requires the application of a wide range of rules and techniques including industrial codes of practice and company standards established from experience. It is partially the totality of these tools that makes each successful enterprise unique and therefore enables it to remain competitive. Therefore, if the strength of the company is to be maintained, let alone, increased by integration, the selection of the necessary software and hardware should be based on a careful analysis of the target functions, especially with regard to their data requirements.

Adequate functional and data analysis is hardly promoted in current trade literature. Instead, many computer manufacturers and systems houses present their integrated products as the total computer solution to all customers' requirements. This is unlikely to be the case in general because there is a limit to which any proprietory software and hardware can meet an individual customer's requirements. The ideal approach recognized in this research is for specialist system vendors to offer 'open' turnkey systems that enabled customers to become productive immediately while, at the same time, allowing future expansion probably with third-party software. The CAE package created in the course of this research work can be viewed as one such system.

9.2.2 **Strategy for implementation**

Having analyzed the functions and data, the next requirement identified is the mapping out of an architecture for the integrated system i.e. the required distribution of the processing power and the data in the integrated system.

Despite the wide range of system architectures made possible by the recent developements in communication standards such as MAP, the analysis carried out in this research regarding the data and its usage in a design system showed a need for its centralized management. This is primarily due to the heterogeneity of the data required by a typical design application. The configuration favoured for integrated systems is therefore one involving a database which can be accessed from anywhere in a design department and by any application. Where implementation on a single multi-user system is not feasible, a hierchical architecture is envisaged where the global database is implemented on a master computer system while maintaining the local databases and associated application programs on a number of lower-level systems distributed on the enterprise's premises. The proposed layering of the system database into a global and local components has been discussed at length in chapter 6.

The main conclusions reached regarding the design of CAE database systems are presented in the next section.

9.3 **The database approach to integration**

In this research, an analysis of the characteristics of engineering design data and activities and also those of discrete mechanical products led to the formulation of a set of specifications and proposals for the design of CAE database systems. A number of general conclusions could be drawn from from this analysis as well as from the experience with the integrated system implemented. The main ones were with regard to (a) the applicability of conventional database practices and technology to the problem of integration in mechanical engineering systems and (b) the optimum modelling of mechanical engineering data. These are presented separately in the subsections that follow.

9.3.1 **Applicability of conventional database concepts and technology**

The development of the database concepts and technology in use today was motivated by the need to manage large volumes of commercial data. This historical background has therefore led to a continuining contention as to their applicability to the management of engineering data. This has been discussed in Chapter 6.

In this research, conventional database tools namely the Entity-Relationship approach to conceptual database design, relational data modelling and a proprietory database management system (dBASE II), were deployed in conjunction with two new concepts i.e. the control module and a two-layered database structure, in a successful implementation of an integrated CAE system for gearbox design. The conclusion could therefore be drawn that it is feasible to apply conventional tools to the task of integration in CAE systems. The ease with which this can be done and the degree of the integration achievable are dependent on many factors including the quality of data analysis and modelling, and the capabilities of the database management system selected.

It is emphasized that the conclusion presented above refers specifically to the problem of integration in CAE systems: it is not a blanket statement about the suitability of contemporary formal database concepts and technology for all other purposes in engineering. Different approaches to data modelling and different DBMS capabilities may well be required to achieve other objectives, such as the enhancement of the performance of an individual application program (where data shareability is no longer the key issue). This possible difference in objectives in the application of database concepts and technology in engineering has been discussed briefly in Sect 6.2.

9.3.2 Analysis and modelling of engineering data for integration

In the application of the database approach to the problem of integration, the optimum analysis and modelling of the data was recognized as being particularly important since these, to a large extent, will determine the performance of the eventual system. Two conclusions could be drawn as follows:-

(a) **Contents of the database for an integrated system.**
Ideally, only truly common data should be modelled in

the global database proposed: the primary requirement in an integrated system is the sharing of common data, not merely that of a common data 'pool'. Restricting the modelling to such data will not only reduce the variety of the data to be considered but will also concentrate the modelling effort on the requirements for integration.

Common data can best be identified by a top-down approach where the data of interest in the CAE system is initially analyzed independently from the data requirements of the various applications in the system. The design of the conceptual database for the gearbox design system showed that the combination of the idea of a hierarchical description of discrete mechanical products and the Entity-Relationship approach to information analysis constitutes a very powerful approach for the recognition of truly common data.

(b) **Use of the relational model of data**

Among the main conventional data models, the relational model is the one that offers the level of flexibility required for integration. This has been discussed at length in Sect 8.4.2. Also, the associated concepts, such as normalization, were seen to be very useful in promoting a more scientific approach to the modelling problem. Relational modelling therefore seems to be particularly suited to the design of databases for integrated CAE systems.

9.4 Suggestions for future research in integrated CAE systems

The increased application of computer technology in engineering and the growing popularity of the concept of computer integrated manufacturing (CIM) have expanded the scope for research in concepts and technology for integration. Since there now seems to be a general concensus that integration does have to involve some form of centralized data management [56], the application of the database concepts in engineering is likely to remain an active area of research for some time. Two research topics of rather immediate interest were identified as follows:-

(a) **Hybrid integrated CAE system architectures**

Whilst a fully centralized architecture has been propounded in this thesis, the developments in communication standards in the last three years mean that it is not difficult to envisage integrated CAE systems of the future being of a hybrid nature i.e. taking advantage of developments in both database and networking technologies. The feasibility of such configurations needs to be investigated.

(b) **Incorporation of artificial intelligence-based expert system features in integrated CAE systems**

The implementation of integrated database systems as proposed in this thesis requires the incorporation of

]

some deductive capacities such as are provided by the control module 'CAECONT' (see Fig 7.3) in the package for gearbox design. It seems possible to expand these capabilities through the use of expert system concepts: research in this area would therefore appear to be of immediate interest.

Other enhancements that might be achieved through the application of such concepts include more intelligent interfacing, query optimization and better maintenance of data consistency in the system. The possibilities of incorporating expert concepts into database systems and vice-versa have been discussed at length by **Jacke** and **Vassiliou** [57]

Apart from these specific topics, there seems to be a need for a deeper study of the process of developing a product from concept through design and manufacture to its marketing as was carried out in the initial stages of this research (see Chapter 3). The results of such studies would not only guide the implementation of integrated systems in individual enterprises but would also provide more solid foundations for the generalized integrated CAE systems of the future.

A P P E N D I X   O N E

The dBASE II RELATIONAL DATABASE MANAGEMENT SYSTEM

A1.1 Introduction

Database management systems (DBMSs) are special software systems which have been developed to aid the storage, manipulation, reporting and managing of data. The basic function of a DBMS is to make data of common interest available to a multiplicity of applications in a computer aided system, while at the same time helping to preserve its integrity and improving the efficiency of the whole data processing function. DBMSs have been discussed in greater detail in Chapter 5, in conjunction with an outline of the main concepts underlying the database approach to information management.

dBASE II is an assembly language database management system produced by Ashton Tate Ltd. For 16-bit machines, it is available for those with CP/M-86 or MSDOS operating systems and requires at least 128K of main memory and one or more mass storage devices, typically floppy disc drives.

A full description of the package is beyond the scope of this appendix: because of its popularity (and that of the more recent product 'dBASE III') there is a good selection of books available, many from third-party sources [59,60,61], in which its operation is described in great detail. The objective in this appendix is to introduce the main features of the package so as to justify its selection in this research work.

A1.2 **dBASE II data files**

dBASE II is based on relational algebra. As has been discussed at length in Chapter 5, the basic construct in relational modelling is the semi-theoretic 'relation' which is a subset of the Cartesian product of a list of domains. At the implementation level, a relation can be viewed as a two-dimensional table with individually-named columns (domains) and where each row (record) is unique i.e. each record can be individually identified. Also, a relationship between two relations is expressed by the presence of a common domain in the corresponding tables.

Accordingly, the basic unit in a dBASE II database system is the database file (called a 'data base' in dBASE II terminology). A picture of the layout of such a file is shown in Fig A1.1. Thus the database structure will be made up of a number of such tables, each with a unique name. Once these have been set up, they are retained on disc and the main function of the DBMS is then to assist the manipulation of the records in them as required by individual user programs

	Fieldname	Fieldname	Fieldname	....
Record No.1	....	....	....	......
Record No.2	....	....	....	......
Record No.3	....	....	....	......
Record No.4	....	....	....	......
Record No.5	....	....	....	......
Record No.6	....	....	....	......
....	....	....	....	......
....	....	....	....	......

Fig A1.1 Showing the structure of a relational data file

or for ad hoc information queries.

The following limitations apply with regard to each dBASE II data file.

Number of fields per file	32 max
Number of characters per record	1000 max
Number of records per file	65535 max
Length of character field	254 max
Accuracy of numeric field	10 digits
Largest numeric entry	1.8×10^{63} approx
Smallest numeric entry	10^{-63} approx

Also, only two data files can be open at any one time, one in each of dBASE II's two 'working areas'.

A1.3 **The dBASE II command language**

The main operations in using any database management system can be classified as follows:-

(1) Creating the datafile structures,
(2) Entering and retrieving data from the database files,
(3) Data manipulation,
(4) Generation of reports involving the data in the database,
(5) Linking to other software.

dBASE II provides english-like commands with which these operations can be performed. Furthermore, one of the most powerful features provided is the ability to expand and tailor the commands by adding phrases that define further what the command should do.

Following is an example of a typical dBASE II command line. (In this Appendix, all dBASE II commands, operators, functions etc., are shown in bold text, any prompts produced by dBASE II are shown in italics and those supplied by the user are shown in plain text)

.LIST FOR uts**>**300 **.AND.** yieldstrss**>INT**(zz)

It is seen to be formed from the following components:-
- a command (e.g.'**LIST**') which expresses what action

is required,

- database field variables (e.g. 'uts' and 'yieldstrss') which are field names in the data files in use,
- memory variables (e.g. 'zz') i.e. variables that are not names of fields in database files,
- constants (e.g. '300'),
- functions (e.g. '**int()**')
- operators (e.g. '**>**')

The example command line given above would list all the records in the database file in use for which the numerical value of ultimate tensile stress (uts) was greater than 300 and that of the yield stress was greater than the integer value of the variable 'zz'.

The main commands with regard to the operations (1) to (5) listed earlier in this section are described in Sects A1. 3.1 to A1.3.5 that follow.

A1.3.1 **Creating the file structures**

The general layout of a relational database file has been described in Sect A 1.2. To create a dBASE II data file, enough information has to be provided to describe each field. This comprises a field name, field type (numeric, character or logical), field length and the number of decimal places in the case of a numeric field.

As an example, consider the creation of a file to hold shaft geometry data. After loading dBASE II, the procedure would be as follows (the text shown in italics is entered by the user while the rest are prompts by the system):-

.CREATE

Enter filename: SHAFT

Enter record structure as follows

Field	*Name*	*Type*	*Width*	*Decomal*
001	SHAFT:NO	C	5	
002	BLANK:DIA	N	8	1

This process would be repeated until the structures of all the fields required have been entered after which data can be stored in the file immediately if so required.

A new data file can also be created with the expression

.CREATE Newfile **FROM** Structurefile

where 'Newfile' is the name of the new file required and 'Structurefile' is the name of a data file which contains all the field information for the new file. This is a very useful feature as it provides the ability to create new data files (usually temporary ones) in a non-interactive manner.

Once a data file has been created, commands are provided for the modification of its structure and for other related operations such as the copying of the structure to a new file.

A 1.3.2 **Entering and retrieving data from the database**

Data can be added to a file either interactively or from another file. The main commands provided are as follows:-

.APPEND which prompts the user for the data he wishes to enter into the various fields in the file in use,

.APPEND FROM otherfile, which adds data from another file into the file in use,

and **.INSERT,** which allows a record to be inserted in the middle of a file.

A number of commands are also provided for locating the required records in a file. Once found, they can either be copied to a different file or the individual field values can be assigned to memory variables as required.

A1.3.3 **Data manipulation**

The ability to manipulate data within a file as well as among several files is the most important feature of any DBMS.

dBASE II provides a rich variety of commands for data manipulation. These fall into the following functional groups:-

- data display (e.g. **BROWSE, LIST, DISPLAY** etc.,)
- data location (e.g. **FIND, LOCATE, SKIP** etc.,)

and - data editing (e.g. **CHANGE, REPLACE, EDIT** etc.,)

The data in a file can also be sorted in a particular order or indexed with the commands

.SORT ON fieldname **TO** newfile

and **.INDEX ON** fieldname **TO** Newfile

respectively. Records in different data files can be manipulated using four major commands as follows:-

.COPY TO newfile and

and **.APPEND FROM** anotherfile

which copy data from and to the file in use respectively,

.JOIN TO newfile **FOR** criteria

which allows records from two files two be joined whenever the specified criteria are met. The new records are appended to a thirdfile. This is one of the most powerful commands in dBASE II as it allows easy assembly of all the data on the database relating to a particular item.

The relational algebra operations SELECTION, PROJECTION and JOIN discussed in Chapter 5 are performed with the dBASE II commands **'LOCATE** (or **'FIND'**), **'COPY'** and **'JOIN'** respectively.

A1.3.4 **Report generation**

Apart from viewing and manipulating the records of interest in a file, it is often required to obtain summaries that include many records which meet certain criteria. dBASE II provides a very powerful facility for report generation. Firstly, the relevant data is assembled in one file using the other manipulation commands reviewed earlier. Then, on issuing the command **'REPORT'**, the user is led through a series of prompts to create a custom format for the report. The information required includes items such as the report title, column headings and widths, which columns are to be totalled etc.,. Once all the information prompted for has been entered by the user, the report is immediately displayed on the VDU and a hardcopy is produced if required. The report format is also automatically saved to disc for the future.

A1.3.5 **Linking to other software**

dBASE II allows interfacing with other software through two mechanisms. Firstly, it can read data from files created by other packages such as high level programming languages and word processors, and can also generate system data files which can be read by other software packages. These operations are carried out with the commands

.APPEND FROM systemfile.**TXT SDF**
and **.COPY TO** systemfile.**TXT SDF**

where 'systemfile' is a standard ASCII text file.

The second mode of interfacing with other software is with the command

.QUIT TO othersoftware

where 'othersoftware' a list of command files that can otherwise be run from the operating system level. These will then be run in series and dBASE can then be re-entered if it is included in the list.

In a database system, such interfacing facilities are very important as they enable the performance of certain functions such as those involving much calculation, which are best done with other packages other than the DBMS.

A1.4 Programming

The commands introduced in the preceding sections can be used individually to perform useful tasks, typically ad hoc information queries. However, for those data management operations which need to be performed frequently, these commands can be assembled in series, thereby constituting extremely powerful dBASE II programs (called 'command files' in dBASE II terminology). The more important programming commands are as follows:-

.DO commandfile
and **.CANCEL** (or **.RETURN**)

which run and cancel a command file respectively,

.DO WHILE ... ENDDO

which allows looping until a terminating condition is satisfied, and

.IF ... ELSE ... ENDIF
and **.DO ... CASE ... ENDCASE**

both of which enable branching in command file execution, depending on specified criteria.

The design of dBASE II programs is similar to that required for the more commonly-used programming languages although, with regard to data processing, the power of an individual dBASE II command line is much greater than that of commands in languages such as BASIC and FORTRAN. Also, dBASE II programming encourages the use of subprograms. As these are stored separately on disc, they can be examined and modified more readily and easily.

A1.5 **Concluding remarks**

The introduction to dBASE II given in this appendix has indicated its many capabilities and has therefore shown why it was selected in the research work. The DBMS's main asset is the powerful data manipulation capability that it offers. Also, the discipline it imposes on the structuring of the data files was found to be very beneficial to the whole data analysis and modelling process for the CAE system's database.

For applications involving frequent access to the database, the interfacing capabilities provided by dBASE II may prove to be unsatisfactory: a DBMS offering host-language interfacing may be preferable. However, in this research work where the use of a DBMS was for purposes of facilitating integration in CAE systems, the interfacing mechanisms offered and the general performance given by dBASE II were found to be quite satisfactory.

A P P E N D I X   T W O

THE APPLICATION PROGRAMS INCORPORATED IN THE GEARBOX DESIGN PACKAGE

The integrated computer aided engineering system for gearbox design implemented in the course of the research work has been described at length in Chapter 7. The discussion in the chapter is concentrated on the design and implementation of the overall package and only brief introductory notes have been given with regard to each of the five application programs included in it. The programs are described in greater detail in this appendix.

The application programs implemented in the CAE system are for (a) gear design, (b) drafting of shafts, (c) bearing selection, (d) shaft design analysis and (e) shaft material specification. These are discussed in Sects (a) to (e) respectively in this appendix. In each section, a brief theoretical foundation for the program is presented firstly, followed by a main flowchart. Sample results given out by each of the programs are shown in Appendix 3 in conjunction with a presentation of two gearbox design examples.

All the programs are coded in Microsoft BASIC except for the one for bearing selection, a part of which is written in the programming language provided by the database management system 'dBASE II' used in the package.

(a) **The gear design program 'GEARDES'**

This is a program for the design of a pair of spur or helical gears conforming to BS 436: 1967. It was originally developed by **Walton D.** and **Taylor S.** (Mechanical Engineering Department, University of Birmingham, 1983).

The design of such gears is based on considerations of the bending and surface strengths of the meshing teeth. The power rating for a gear pair based on bending and surface strength respectively, can be obtained using the following equations:-

$$P = (X_b S_b Y F N T m^2)/(19.1 \times 10^6) \quad \text{---- (1)}$$

and

$$P = (X_c S_c Z F N T m\text{^}{}^{1.8})/(19.1 \times 10^6) \quad \text{---- (2)}$$

where P = power rating (kW)

X_b = speed factor for strength, which takes into account the effect of the operating speed on the nominal bending strength of a gear tooth,

X_c = speed factor for wear, which takes into account the effect of the operating speed on the wear resistance of the tooth flank,

Y = strength factor, which accounts for the effect of the tooth form on the nominal

Z = zone factor, which takes into account the influence of the relative curvature of the teeth flanks on the Hertzian stresses,

S_b = bending strength rating for the gear material (N/mm^2),

S_c = surface strength rating for the gear material (N/mm^1.8)

F = gear facewidth (mm)

N = gear speed (rev/min)

m = module (mm)

T = number of gear teeth.

More up-to-date design equations are given in BS 436: Part III: 1986: 'Methods for calculations of contact and root bending stress limitations for metallic involute gears'. These equations take into account many more strength reduction factors.

Charts are available for determining the values of the speed, strength and zone factors. Guidelines are also provided for obtaining the number of gear teeth to avoid undercutting, a first approximation of the facewidth and the appropriate module (the adoption of a standard module is advantageous as cutters are readily available for the machining of such gears). Once these values have been obtained, the required strength and wear ratings for the pinion material can then be calculated using equations (1) and (2) thereby enabling the selection of a standard gear material. The selection will selection of a standard gear material. The selection will generally be determined by the wear properties of the material rather than its bending strength. The same procedure is used to select the material for the wheel. To avoid overdesigning, the strength and wear ratings for the selected materials can be substituted into equations (1) and (2) to determine the optimum facewidth. The largest facewidth calculated for either the pinion or the wheel is then adopted for the gear pair.

The design procedure outlined is very iterative and the extra considerations in the design may include the overload capacity, operating noise levels, resonance and the limiting ruling sections of the various standard gear materials.

The main flowchart for the program is shown in Fig A2.1 while samples of the output from it are indicated in Appendix 3.

The details of the two menus highlighted in the flowchart are as follows:-

Menu at start	Menu at end
List gear specs.	List gear specs.
Change the specs.	Gear design results
Design the gear pair	Display design details
Notes on printer	Print specifications & design
Stop the program	Material change
	Change module
	Increase teeth
	Extend centre distance
	Redesign, revise specifications
	Keep results on database
	Notes on printer
	Stop design without saving

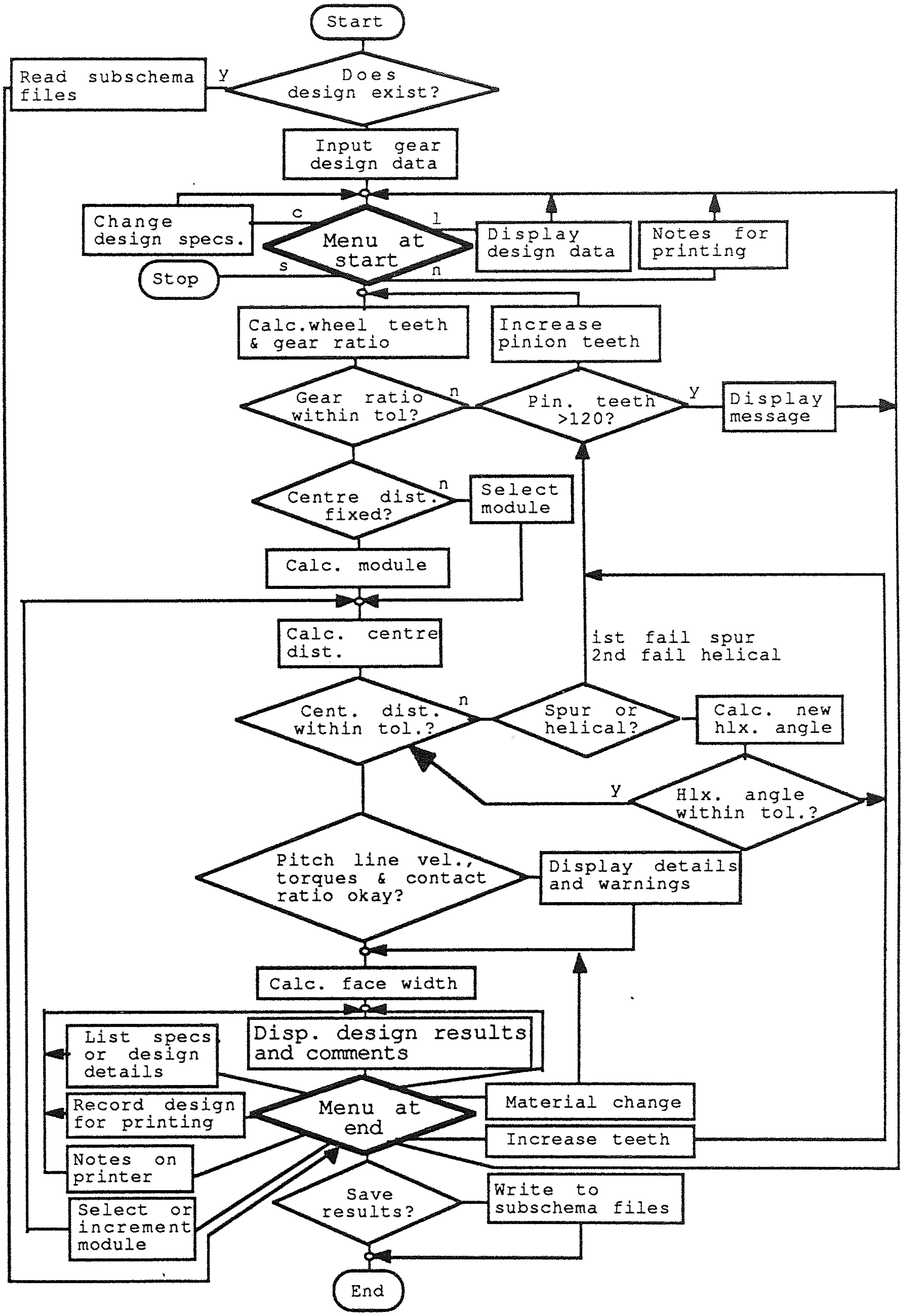


Fig A2.1 Showing the flowchart for the gear design program GEARDES

(b) **The drafting program DRAW1**

This program allows the drawing of a completely axisymmetric shaft. It was specially created for purposes of the research and is therefore not a generalized drafting program. The main flowchart for the program is shown in Fig A2.2.

As the program is coded in MSBASIC which has no graphics commands per se, the program utilizes the Graphics System Extension (GSX) of the MSDOS operating system to effect graphics on the VDU. GSX incorporates graphic capabilities into the operating system and provides a host language interface for application programs written in most high level languages that support the GSX calling conventions.

Briefly, the main benefits provided by GSX as follows:-

- It provides graphics primitives (e.g. circle, arcs,bars, filling etc.,) for implementing graphics applications with reduced programming effort.
- GSX provides a device-independent software interface for application programs i.e. a GSX graphics program can work with different output devices such as printers and plotters. this is achieved through the use of individual GSX device drivers which translate the standard GSX calls issued from the program to the characteristics of a particular device.
- Any graphics application program that uses GSX can be run

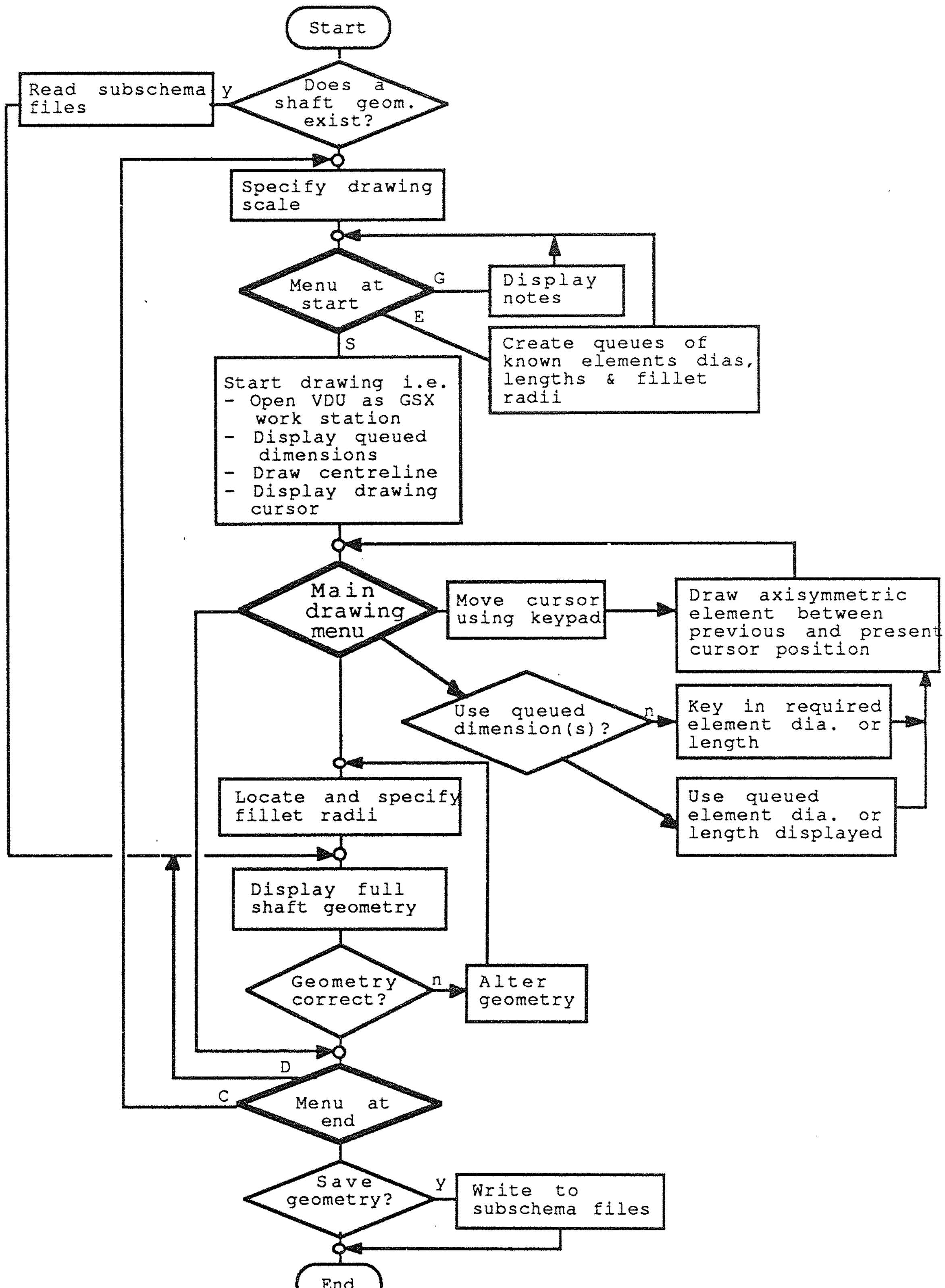


Fig A2.2 Showing the flowchart for the drafting program DRAW1 with several 8080 and 8086 microcomputers operating systems such as MSDOS, CP/M and CP/M-86.

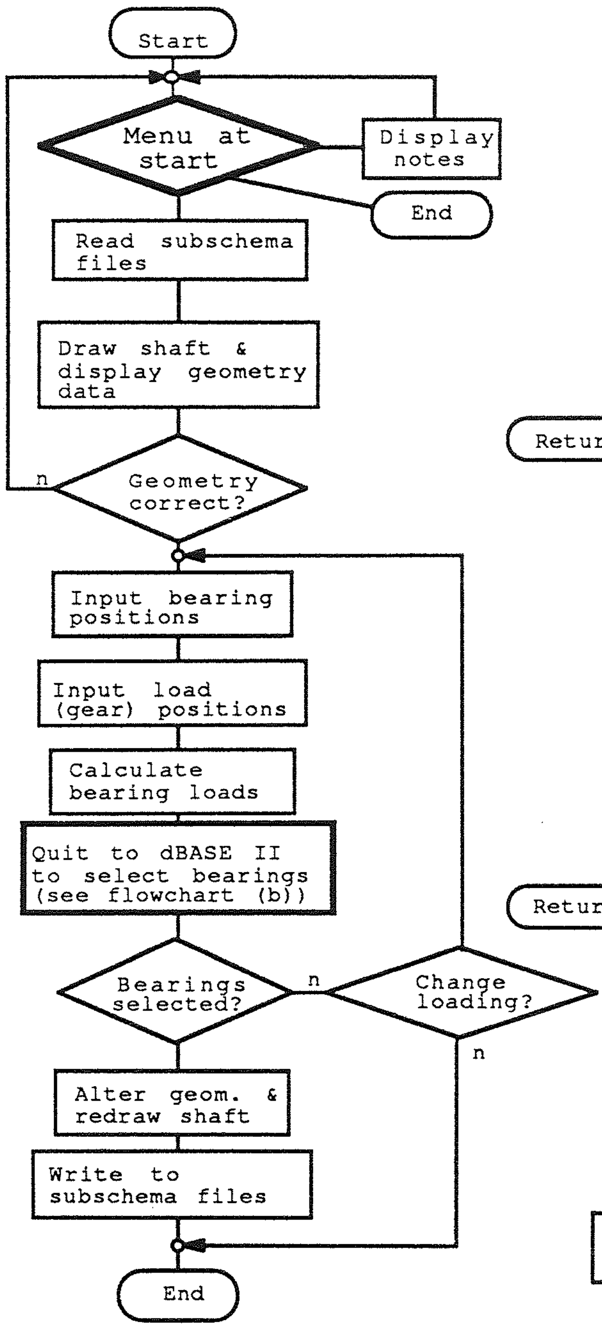
(Detailed information on GSX is given in the publications 'GSX-86 User's Guide' and 'GSX Programmer's Guide' published by Digital Research Inc., 1984)

The output from the program consists of (a) the geometry i.e. the length and the end diameters of the elements that make up the shaft and (b) the fillet radii on the shaft, if any. These are stored in the database as shaft geometry data. Sample outputs are indicated in Appendix 3.

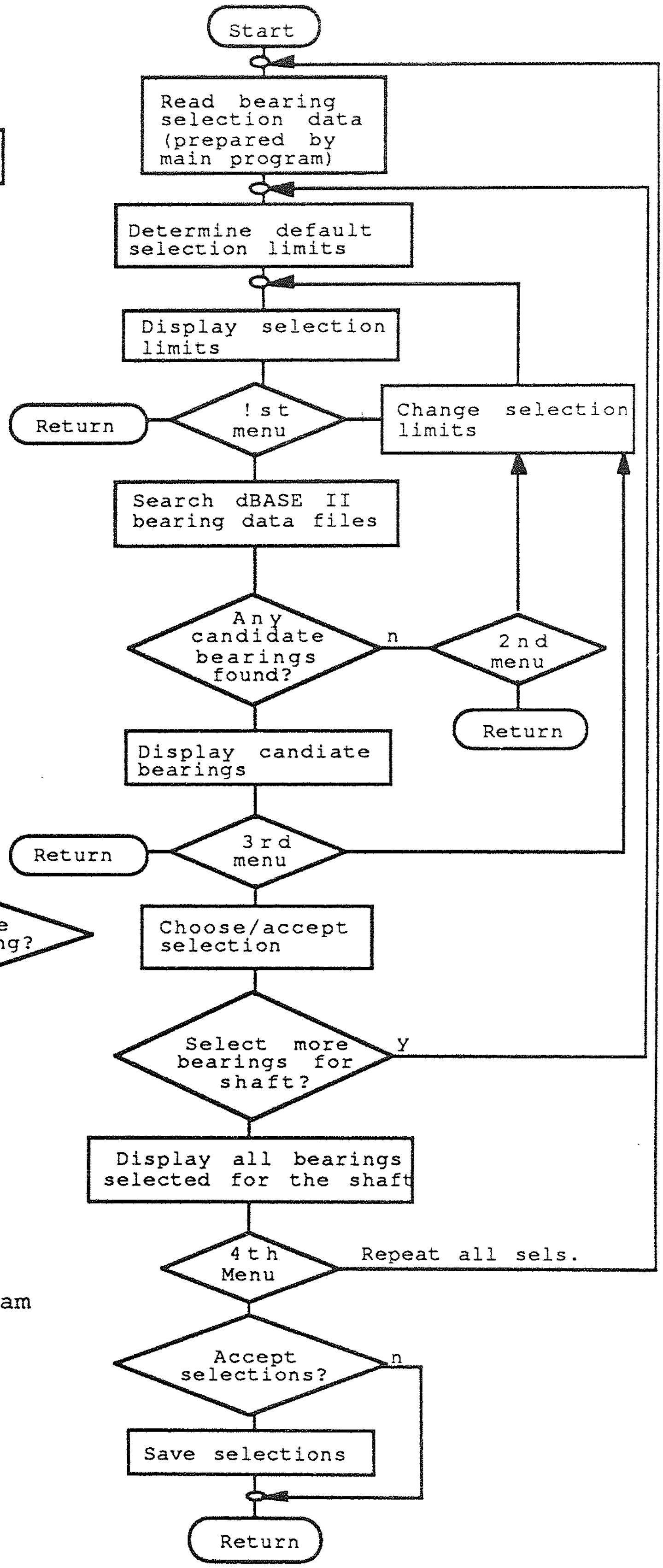
(c) **The bearing selection program BSELECT**

This program was created in the course of the research work to allow the selection of appropriate bearings for a gearbox. As the bearings data was stored in dBASE II data files and the selection process required repetitive access to this data, the data preparation part of the program is coded in MSBASIC while the part for actual selection is implemented in the dBASE II programming language. The flowcharts for the two parts of the program are shown in Fig A3.1.

The program is based on the guidelines given by the SKF Group for selection from their range of rolling contact bearings (see SKF General Catalogue 3000 III E/GB 666 III, June 1980).



(a) Flowchart for main program in MSBASIC



(b) Flowchart for subprogram in dBASE II programming language

Fig A2.3 Showing flowcharts for the bearing selection program BSELECT

The bearing selection process can be described to consist of two stages i.e. the selection of (a) a bearing type and (b) a bearing size. These stages are briefly discussed below.

(1) Selection of bearing type.

A large range of bearing types is available, each with characteristic features which make it particularly suitable for certain applications. Many factors must be considered when choosing a bearing type, the main ones being the magnitude and directions of the bearing loads, angular misalignment, running speed, available space, running accuracy and noise level. The characteristics of the various bearing types are well documented in the bearing catalogues.

(2) Selection of bearing size.

The selection of a bearing size is made on the basis of its load carrying capacity and the requirements with regard to life and reliability. with a knowledge of the loads and the required life of the bearing, the load rating for the appropriate bearing can be obtained from the equation

$$C = (L_{10})^{1/p}P \qquad --- (3)$$

where C = the required basic dynamic load rating (N). This in the load for which 90% of a sufficiently large group of a particular bearing will run for at least a million revolutions before flaking occurs on one of the rings or rolling elements. This load is specified for each bearing listed in the catalogue.

L_{10} = the basic rating life i.e. the required life of the bearing, expressed in millions of revolutions.

P = the equivalent dynamic bearing load (N). This is an hypothetical load, constant in magnitude and direction, acting radially on a radial bearing, which, if applied, would have the same influence on the life of the bearing as the combination of the actual axial and radial loads acting.

p = an index (3 and 10/3 for ball and roller bearings respectively).

Guidelines are given in bearing catalogues regarding the calculation of the equivalent dynamic bearing loads and the typical rating lives for bearings in different applications (10000 to 25000 hrs for general purpose gear drives), including various modifying factors for the effects of required reliability, operating conditions etc.,

In using the program BSELECT, the selection is limited to Self-aligning ball, spherical roller and taper roller bearings and no modifying factors are applied to either basic equivalent dynamic load calculated or the specified bearing rating life. These limitations came about purely for pragmatic reasons i.e. program simplicity and the reduction of the amount of bearing data to be implemented in the CAE database. Also, once the bearing selections have been accepted by the designer, the shaft geometry is automatically modified to suit.

(d) **The shaft analysis program STATIC**

This is a chain of programs for the static analysis of a shaft as a bar, a torsional member and a beam. The separate programs incorporated were originally developed in the department as teaching aids in stress and vibration analysis.

The programs are based on simplified finite element analysis. This is a numerical method for obtaining approximate solutions to many types of engineering problems. The need for such methods arises from the fact that while the governing equations and boundary conditions can be obtained for many practical problems in engineering, their analytical solution may be extremely difficult or even impossible due to problems introduced by irregular geometry or other discontinuities. In the finite element approach, the region of interest is divided up into a number of connected elements within which approximate functions (also called interpolation models) are used to represent the field variable. These models are defined in terms of the values of the field variables at the element nodes. Once the nodal values are known after solving the field equations for the whole continuum, the interpolation models define the field variable throughout the assemblage of elements.

Following is an outline of the finite element approach as used in the static analysis programs.

(1) Discretization of the shaft.

Since a shaft can generally be taken to be a one-dimensional structure, it is convenient to define nodes at all shoulders and at bearing andgear positions.

(2) Selection of interpolation models for displacements.

In general, the interpolation functions are taken to be polynomials because it is easier to formulate and computerize the resulting finite element equations and it is also possible to improve the accuracy of the results by increasing the number of terms in the polynomial.

The models used in the programs are as follows:-

$$q(x) = A + Bx \quad \text{for tension and torsion.}$$

$$\text{and } q(x) = A + Bx + Cx^2 + Dx^3 \quad \text{for bending.}$$

(3) Derivation of element stiffness matrices.

The expression for the strain energy of an individual element can be written in the form

$$U^{(e)} = 1/2\{u\}^{t}[K]^{(e)}\{u\}$$

where $\{u\}$ and $[K]^{(e)}$ are the element displacement vector and stiffness matrix respectively.

The interpolation models used gave the following stiffness matrices.

$$[K]^{(e)} = \frac{A^{(e)}E^{(e)}}{L^{(e)}}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{for tension}$$

$$[K]^{(e)} = \frac{G^{(e)}J^{(e)}}{L^{(e)}}\begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} \quad \text{for torsion}$$

and

$$[K]^{(e)} = \frac{2E^{(e)}I^{(e)}}{(L^3)^{(e)}}\begin{bmatrix} 6 & 3L & -6 & 3L \\ 3L & 2L^2 & -3L & L^2 \\ -6 & -3L & 6 & -3L \\ 3L & L^2 & -3L & 2L^2 \end{bmatrix} \quad \text{for bending}$$

(4) Assembly of element stiffness matrices and loadvectors.

In the programs, the stiffness matrices and load vectors are assembled in a suitable manner to obtain the overall equilibrium matrix equation

$$[K]\{q\} = \{P\}$$

where [K] is the stiffness matrix for the whole shaft, {q} is the vector of nodal displacements and {P} is the vector of nodal forces.

(5) Solution for the unknown nodal displacements.
After modifying the overall equilibrium equation to take into account the boundary conditions of the problem, it can then be solved to obtain the the nodal displacement vector {q}. These values can then be used in the various interpolation models to approximate the displacement at any point on the shaft.

(6) Computation of element strains and stresses.
The nodal displacements obtained can then be used in conjunction with the interpolation models to obtain the element strains and stresses by applying the appropriate equations from solid mechanics.

The general flowchart for the chain of programs STATIC is shown in Fig A2.4. Example outputs from the set of programs are given in Appendix 3.

(e) **The material specification program MATSPEC**

This program was specially created in the research work to

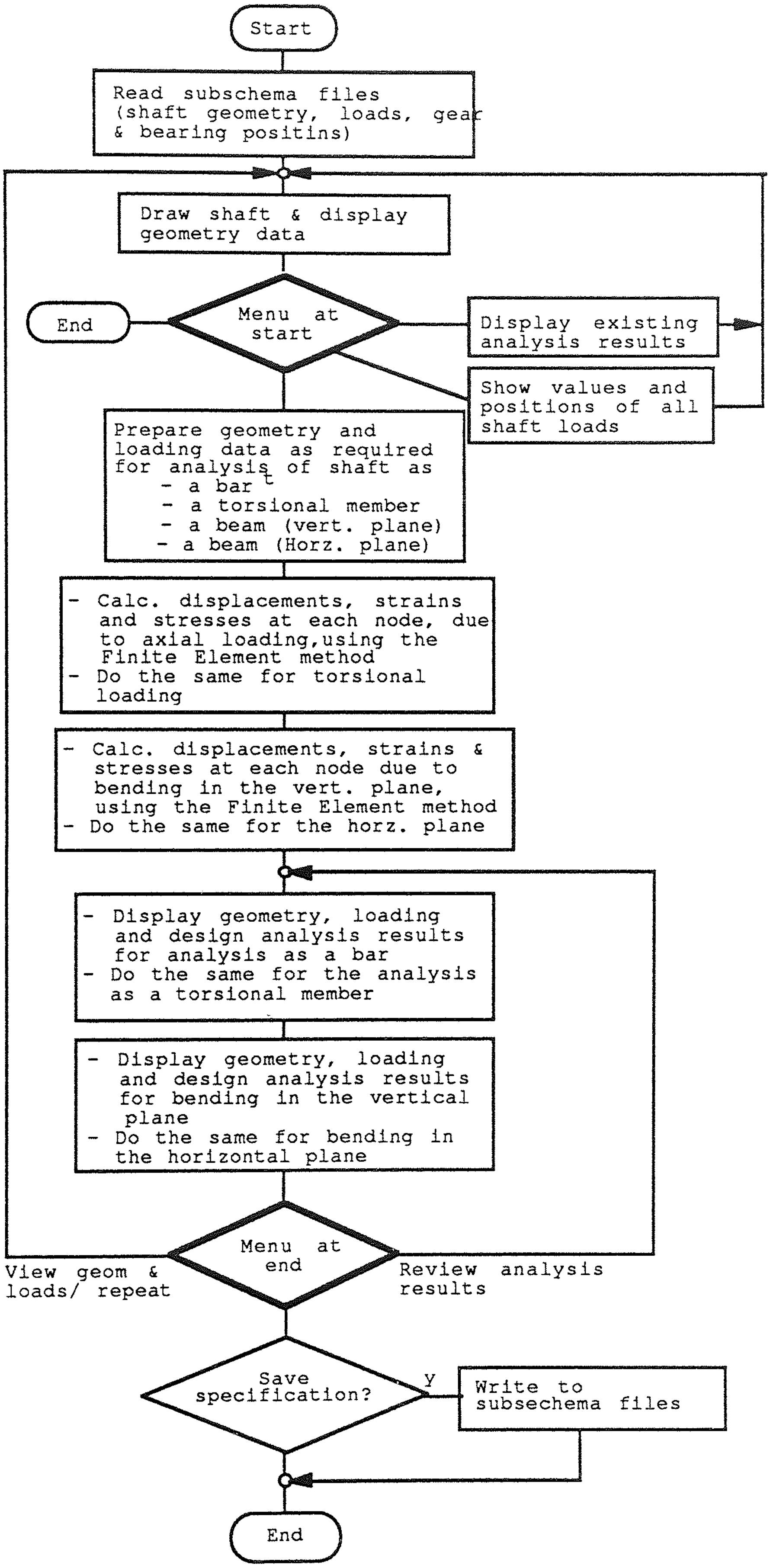


Fig A2.4 Showing the main flowchart for the static analysis program STATIC enable the specification of the properties of an appropriate material for the shaft. The program produces an expression of the form

$$N = f(S_y, S_u) \qquad ---- (4)$$

where N is the required safety factor and S_y and S_u are the yield and ultimate tensile strengths respectively, of the appropriate material. This expression can then be used in conjunction with shaft material data to select a material for the shaft.

The program is based on the use of the Soderberg method for assessment of the fatigue safety factor for a machine part, as discussed by Peterson [**58**]. For a shaft in combined tension, torsion and bending the safety factor can be obtained from the expression

$$1/N = ([\sigma_{od}/S_y + \sigma_{ob}/L_b S_y + K_{tf}\sigma_a/S_f]^2 + 3[\tau_o/L_s S_y + K_{tfs}\,\tau_a/S_f]^2)^{1/2} \qquad --- (5)$$

where σ_{od} = steady normal stress (kN/mm^2)

S_y = yield strength (kN/mm^2)

σ_{ob} = steady bending stress (kN/mm^2)

L_b = limit design factor for bending (1.7 for a round bar)

K_{tf} = estimated fatigue notch factor for normal stress (=$q(K_t-1)+1$ where q is the notch sensitivity and K_t is the theoretical stress concentration factor for normal stress),

σ_a = alternating normal stress (kN/mm^2)

S_f = fatigue strength for axial loading or bending of an unnotched specimen (kN/mm^2)

τ_o = steady or static shear stress (kN/mm^2)

L_s = limit design factor for torsion (4/3 for a round bar)

K_{tfs} = estimated fatigue notch factor for shear stress (=$q(K_{ts}-1)+1$ where K_{ts} is the theoretical stress concentration factor for shear stress)

τ_a = alternating shear stress (kN/mm^2).

The shafts in a gearbox can be considered to have a steady normal stress (from axial loading), a steady shear stress (from the torque applied) and a reversed bending stress (from a combination of radial loading and rotation) only. These are obtained from shaft analysis results stored in the database.

In the design of the program, it was assumed that it would not generally be possible to obtain the figures for the fatigue strength S_f of materials directly from literature. The large part of the program is therefore devoted to the determination of an expression for the fatigue strength S_f. Shigley [63] gives the following expression for determining the fatigue strength of a ductile material from its ultimate tensile strength.

$$S_f = K_a K_b K_c K_d K_f S_e'$$

where K_a = surface factor

K_b = size factor

K_c = reliability factor

K_d = temperature factor

K_f = miscellaneous effects factor

and S_e' = endurance limit of rotating beam specimen (kN/mm^2). This should generally be obtained by experiment but Shigley gives $S_e' = S_u/2$ as a general approximation.

When using the program, the values of the various fatigue strength reduction factors K_a to K_f are either entered by the user or default values are used.

Assuming full notch sensitivity (i.e. q=1 in the expression

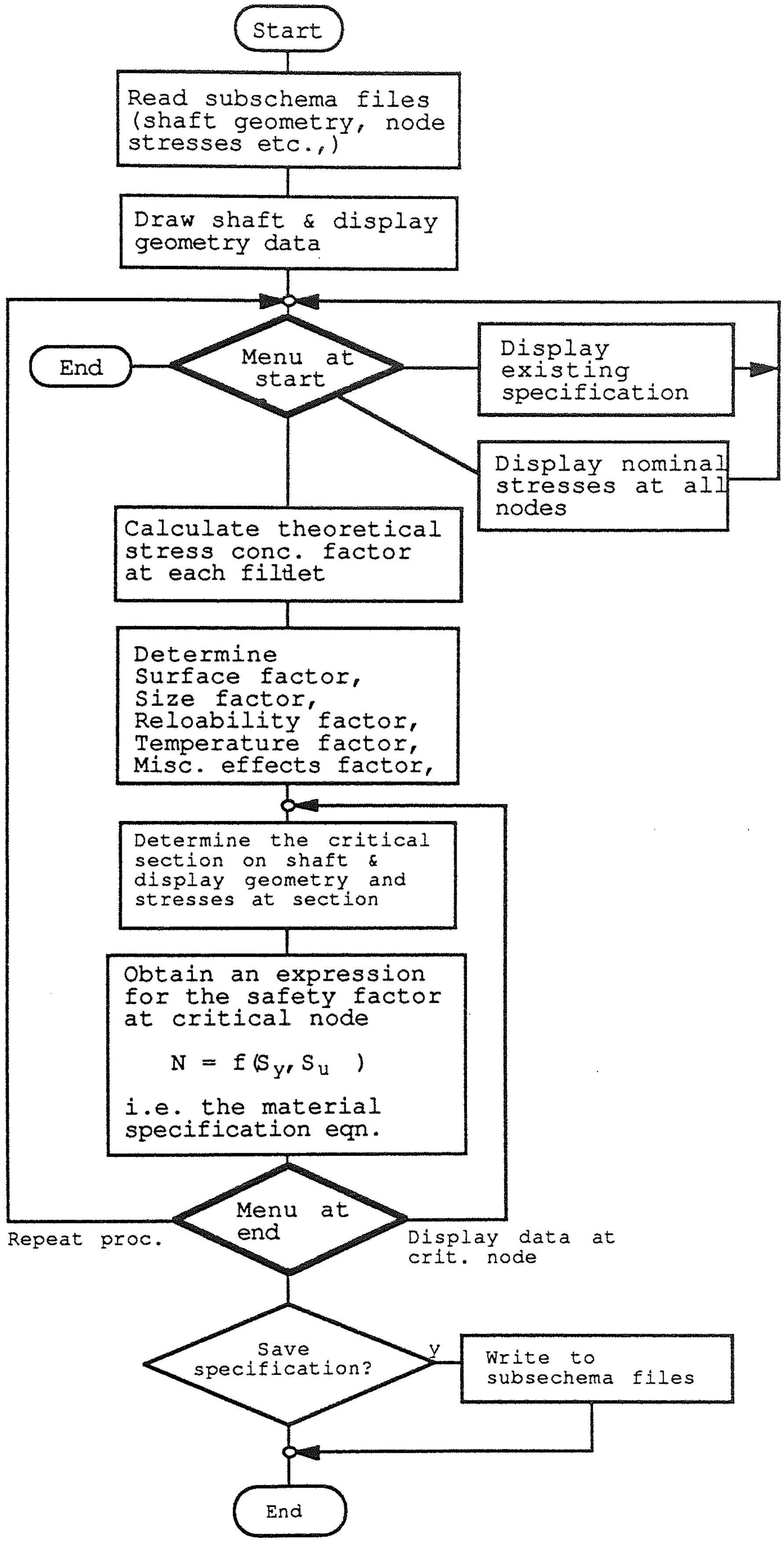


Fig A2.5 Showing the main flowchart for the material specification program MSELECT

$K_{tf} = q(K_t-1)+1)$ then, the fatigue notch factor K_{tf} can be estimated to equal the theoretical stress concentration factor K_t for normal stress. Its values at various nodes on the shaft can therefore be obtained from polynomial functions obtained earlier in the research for the stress concentration curves given by Peterson [**58**].

By substituting for all known values in equation (5), a material specification equation of the form (4) is obtained for the critical node. By substituting for values for Sy and Su each candidate material, the expression can then be used to select the one that would give the required safety factor.

The flowchart for the program is given shown in Fig A3.5. Sample outputs from the program are given in Appendix 3.

A P P E N D I X   T H R E E

USING THE INTEGRATED CAE PACKAGE: A GEARBOX DESIGN EXAMPLE

A3.1 Introduction

The design of an integrated computer aided engineering system for the design of industrial gearboxes has been presented in Chapter 7. Also, the theoretical basis and the main flowchart for each of the five application programs incorporated in the package have been discussed at length in Appendix 2. A gearbox design example is presented in this appendix in order to (a) indicate the use of the package in general and the role of the CAE control module (described in Chapters 6 and 7) in particular and (b) show examples of the output from each of the programs in the system.

The scope of the package created has been discussed at in Chapter 8. The function facilitated by the system are as follows:-

(a) Job initiation.
This involves the generation of an identification code and the selection of a standard configuration for the particular gearbox design (job) being initiated. Note that the package has been designed such that a number

of jobs can be handled concurrently, and each of them can be at any stage of the design process; thus the necessity to attach an identifier to each job at the very beginning of the design.

(b) Gear design.

(c) Drafting of shafts.

(d) Bearing selection.

(e) Static analysis of shafts.

(f) Material specification for shafts.

The correct sequencing of these various functions is ensured by the control module. For example, although the gear pairs in a gearbox need not be designed consecutively, the design of a gear located on a particular shaft must be completed before bearings for that shaft can be selected.

In this appendix, the main menus produced by the CAE control module are shown at appropriate stages, in addition to the results given by each of the application programs. Also, although a full design of a typical gearbox requires repeated use of many of the application programs (because of the duplication of similar elements such as shafts, in the gearbox configuration), only one set of results from each program is shown in the example presented. This is partly in order to minimize the length of the appendix and also avoid possible repetition.

With the computing facilities used i.e. Apricot Xi computer system and an Epson FX80 printer, it was not possible within the scope of the research work to arrange for the provision of hardcopies of all the results given out by the package. For purposes of this thesis therefore, menus and results were manually copied from the screen and subsequently formatted to resemble, as closely as possible, the corresponding VDU displays.

For clarity of description, all user entries and the resulting selections are shown in italics in the screen menus indicated in this appendix.

A3.2 **A gearbox design example**

The following main design specifications were assumed for the gearbox design example outlined in this appendix.

Power to be transmitted	20 kW
Input speed	1200 rpm
Speed reduction	4:1
Centre distance	250 mm
Duty	12 hrs/day

(a) **Job initiation**

On starting the integrated CAE system (by entering 'CAECONT' at the MSDOS system level in the directory '\GEARBOX'), the system menu appeared as shown overleaf.

Integrated Computer Aided Engineering System

INDUSTRIAL GEARBOX DESIGN

System level

Explanatory notes

Continue with an existing job

Initiate or cancel a job

Quit integrated CAE system

Your choice? *I*

Selecting the option shown produced a prompt for a job code as follows.

INDUSTRIAL GEARBOX DESIGN

Job initiation

The job number is of the form xxxN where xxx is the job code and N stands for the nth job under the job code

Job Code	Serial No.	Date
EMP	7	23/02/87

The job identifier generated is EMP7

Menu: *Confirm entry*
Repeat entry
Abort job initiation

Enter your choice? *C*

After confirming the job identifier as 'EMP7', the following prompt for the selection of a standard gearbox configuration then appeared.

INDUSTRIAL GEARBOX DESIGN

Job No EMP7

SELECTION OF A STANDARD GEARBOX CONFIGURATION

STANDARD ARRANGEMENTS:

Configuration No.1	Configuration No.2
Single reduction	Double reduction
2 Spur gears (G1 & G2)	4 Helical gears (G1 to G4)
1 gear mesh (G1:G2)	2 gear meshes (G1:G2 & G3:G4)
2 parallel shafts (S1 & S2)	3 parallel shafts (S1 to S3)
4 self-aligning ball brgs (B1 to B4)	6 taper roller brgs (B1 to B6)

Menu: *Specify required configuration*
View standard configurations
Quit job initiation

Enter your choice? *S*

Selecting the second option on the menu allowed the viewing of the standard gearbox configurations presently supported by the system (see Fig 7.5). Selecting the first option as shown allowed the specification of a configuration for the gearbox i.e. configuration No.1. This completed the job initiation procedure.

(b) **Gear design**

Following the completion of the job initiation procedure, the job menu was immediately displayed as follows.

INDUSTRIAL GEARBOX DESIGN

Job level

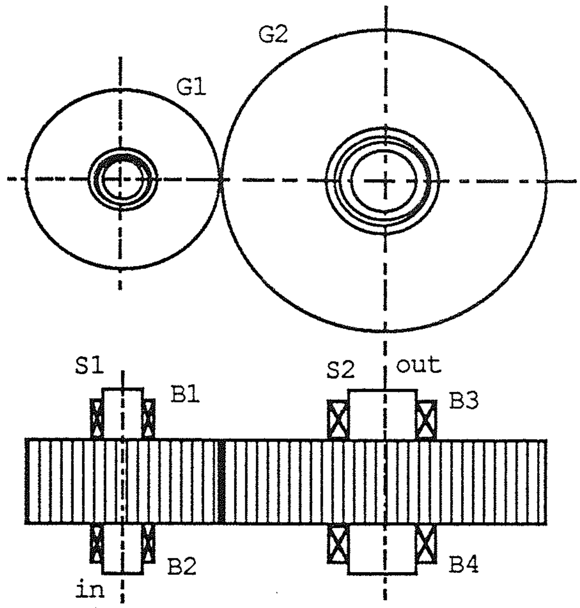
Job Number EMP7

Configuration: No.1
Date started: 23/02/87

Job module	State
Gear pair design	
Draw shaft	
Bearing Selection	
Shaft analysis	
Material specification	

Enter the required option (G, , , ,) *V*
OR enter V=view configuration, Q=quit this job

Selecting the option to view the gearbox configuration caused a display of the configuration for the job as shown below.



CONFIGURATION No.1

Single reduction
2 spur gears (G1 & G2)
2 parallel shafts (S1 & S2)
4 self-aligning ball bearings (B1 to B4)

Selecting the option to design the gear pair caused a corresponding application menu to be presented as shown below.

INDUSTRIAL GEARBOX DESIGN

Application level

Job Number EMP7

Configuration: No.1
Date started: 23/02/87

Application: **Gear design**

Function module	State
Design gear pair No.1	

Enter the required option (1, ,) *1*
OR enter 'Q' to discontinue with this application

Note that only one gear pair was indicated on the menu as this is the only one in the configuration chosen for the job. Choosing to continue with the design as shown caused the system to load and run the gear design program GEARDES.

The main gearbox design specifications listed at the beginning of this section were used as input to the program. The results obtained are listed below:-

GEAR SPECIFICATIONS

Gear ratio	4	Lower tol .2	Upper tol. .2
Minimum teeth		16	
Centre dist (mm)	250	Lower tol. 5	Upper tol. 10
Spur gears			
Power (kW)		20	
Pinion speed (rpm)		1200	
Duty (hrs/day)		12	
Materials:-		Pinion	Gear
Material No.		6	5
UTS (N/mm^2)		540.6	617.8
Surf. stress ($N/mm^{1.8}$)		19.31	13.79
Bend. stress (N/mm^2)		117.2	168.9

Type: cut, 20 deg. pressure angle.
Module: First preference sizes.

GEAR DESIGN RESULTS

Pinion teeth: 17	Wheel teeth: 68
Normal module (mm) 6	
Gear ratio: 4	Error: 0
Spur gears	
Centre dist (mm): 255	Error: 5

Face width (mm): 46.9 (=7.82 x module)
Suggested FW (mm) 46.9 to 78
Wheel wear governs facewidth
Facewidth reasonable

DESIGN DETAILS

	Pinion		Wheel
Material No.	6		5
Number of teeth	17		68
Pitch circle dia (mm)	102		408
Outside dia (mm)	117.6		416.4
Addendum (mm)	7.8		4.2
Profile shift (mm)	1.8		-1.8
Root dia (mm)	90.6		389.4
Base circle dia (mm)	95.85		383.39
Pressure angle (deg)		20	
Speed (revs/min)	1200		300
Safety factor	33.15		31.9
Torque (Nm)	159.15		636.62
Contact ratio		1.586	
Pitch line vel (m/s)		6.4	
Tangential force (N)		3120.7	
Radial force (N)		1135.8	
Axial force (N)		0	

On choosing to accept the gear design results listed, these were saved on the database and the gear design function menu was subsequently redisplayed as shown.

INDUSTRIAL GEARBOX DESIGN

Application level

Job Number EMP7

Configuration: No.1
Date started: 23/02/87

Application: **Gear design**

Function module	State
Design gear pair No.1	**Completed**

Enter the required option (1, ,) *Q*
OR enter 'Q' to discontinue with this application

Note that it was possible at this stage to choose to run the gear design program again in order to either (a) alter the design or (b) review the design results. In this case it was decided to proceed with the subsequent phases of the design.

(b) **Drafting of shafts**

Having selected to leave the gear design function, the following job menu was presented.

INDUSTRIAL GEARBOX DESIGN

Job level

Job Number EMP7

Configuration: No.1
Date started: 23/02/87

Job module	State
Gear pair design	**Completed**
Draw shaft	
Bearing Selection	
Shaft analysis	
Material specification	

Enter the required option (G,D, , ,) *D*
OR enter V=view configuration, Q=quit this job

Note that the gear design module was shown as having been completed because the gear pair just designed is the only one in the standard configuration chosen for the job.

Selecting to start the design of shafts as shown caused the following function menu to be displayed. The menu shows that it was possible at this stage to choose to draw either one of the shafts in the gearbox because this function does not require any input data to be created previously.

INDUSTRIAL GEARBOX DESIGN

Application level

Job Number EMP7

Configuration: No.1
Date started: 23/02/87

Application: **Shaft design (drawing)**

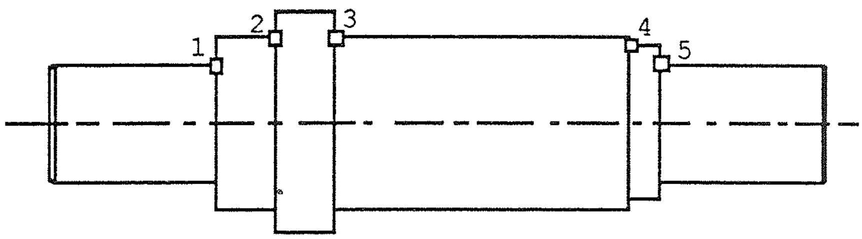
Function module	State
Design (draw) shaft No.1	
Design (draw) shaft No.2	

Enter the required option (1,2,) *1*
OR enter 'Q' to discontinue with this application

Selecting to proceed with the design of shaft No.1 as shown above caused the system to activate the drafting program DRAW1. The final shaft geometry created using the program is shown below.

FILLETS

No.	Radius
1	0.25
2	0.5
3	0.5
4	0.5
5	0.25



Element geometry (Elements numbered from left-hand side)

Elem No.	1	2	3	4	5	6	7	8
Length (mm)	1	20.7	10	10	49.5	5	20.6	1
Left dia (mm)	32.8	37.2	53.6	70.2	53.6	47	37.2	37.2
Right dia (mm)	37.2	37.2	53.6	70.2	53.6	47	37.2	32.8

On choosing to accept this initial shaft geometry, the drafting function menu was redisplayed as shown overleaf, after saving to the database.

INDUSTRIAL GEARBOX DESIGN
Application level

Job Number EMP7   Configuration: No.1
Date started: 23/02/87
Application: **Shaft design (drawing)**

Function module	State
Design (draw) shaft No.1	**Completed**
Design (draw) shaft No.2	

Enter the required option (1,2,) *Q*
OR enter 'Q' to discontinue with this application

(c) **Bearing selection**

Having selected to leave the drafting function as shown above, the job menu was then displayed as shown.

INDUSTRIAL GEARBOX DESIGN
Job level

Job Number EMP7   Configuration: No.1
Date started: 23/02/87

Job module	State
Gear pair design	**Completed**
Draw shaft	**Incomplete**
Bearing Selection	
Shaft analysis	
Material specification	

Enter the required option (G,D,B, ,) *B*
OR enter V=view configuration, Q=quit this job

Note that the drafting function was shown to be incomplete because the geometry of the second shaft in the gearbox configuration had not yet been created at this stage.

Selecting to go on to the bearing selection application as shown above caused the corresponding function menu shown overleaf to be displayed.

INDUSTRIAL GEARBOX DESIGN

Application level

Job Number EMP7 — Configuration: No.1

Date started: 23/02/87

Application: **Rolling bearing selection**

Function module	State
Select bearings for shaft No.1	
Select bearings for shaft No.2	

Enter the required option (1, ,) *1*

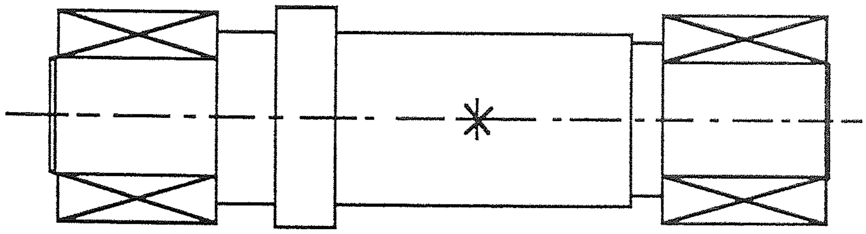
OR enter 'Q' to discontinue with this application

The menu provided only one option as the selection of bearings requires the prior creation of a shaft geometry.

After selecting the option to proceed with bearing selection as shown in the menu above, the program BSELECT was activated. The main results obtained with the program are summarized below.

Bearing Loads

After prompting for the locations of the bearings and the one gear on the shaft, the program calculated and displayed the bearing loads as shown overleaf.



Bearing position	Elem 2	Elem 7
Axial Load (N)	0	0
Vertical load (N)	1317.7	1816.1
Horizontal load (N)	-479.7	-661.1

After prompting for the required bearing life (13000hrs), the program then created default selection limits which could be altered by the user in the process of selecting appropriate bearings. The parameters and limits used in making the final bearing selections are listed below.

SELECTION LIMITS

For bearing on element No.2
Required type: 'Self-aligning ball bearings'

	Specified value	Upper limit	Lower limit
Dyn. load rating (N)	13717	13717	27434
Limiting speed (rpm)	1200	1200	10000
Bearing bore dia (mm)	37.2	33.48	40.92
Bearing width (mm)	20.7	18.63	22.77

For bearing on element No.7
Required type: 'Self-aligning ball bearings'

	Specified value	Upper limit	Lower limit
Dyn. load rating (N)	18905	18905	37810
Limiting speed (rpm)	1200	1200	10000
Bearing bore dia (mm)	37.2	33.48	40.92
Bearing width (mm)	20.7	18.63	22.77

The 'SKF' bearings eventually selected were then displayed as follows:-

Bearing position	Type of bearing	SKF brg Number	Bore dia (mm)	Brg width (mm)	Min abut. (mm)
Elem 2	Slf-align ball	1307	35	21	43
Elem 2	Slf-align ball	1307	35	21	43

Accepting these selections caused an automatic modification of the shaft geometry to suit. The final geomtry for the shaft was displayed as shown below and saved on the database together with the details of the selected bearings.

Elem No.	1	2	3	4	5	6	7	8
Length (mm)	1	21	10	10	49.5	5	21	1
Left dia (mm)	32.8	35	53.6	70.2	53.6	47	35	35
Right dia (mm)	35	35	53.6	70.2	53.6	47	35	32.8

Having accepted the selections made, the bearing selection function menu was then redisplayed as follows.

INDUSTRIAL GEARBOX DESIGN

Application level

Job Number EMP7 — Configuration: No.1

Date started: 23/02/87

Application: **Rolling bearing selection**

Function module	State
Select bearings for shaft No.1	**Completed**
Select bearings for shaft No.2	

Enter the required option (1, ,) *Q*

OR enter 'Q' to discontinue with this application

(d) **Static analysis of shafts**

Choosing to leave the bearing selection application as shown above caused presentation of the job menu shown below.

INDUSTRIAL GEARBOX DESIGN

Job level

Job Number EMP7

Configuration: No.1
Date started: 23/02/87

Job module	State
Gear pair design	**Completed**
Draw shaft	**Incomplete**
Bearing Selection	**Incomplete**
Shaft analysis	
Material specification	

Enter the required option (G,D,B,S,) *S*
OR enter V=view configuration, Q=quit this job

After choosing to proceed to the static analysis of the shaft, the corresponding function menu shown below was then presented.

INDUSTRIAL GEARBOX DESIGN

Application level

Job Number EMP7

Configuration: No.1
Date started: 23/02/87

Application: **Shaft Analysis**

Function module	State
Design analysis of shaft No.1	
Design analysis of shaft No.2	

Enter the required option (1, ,) *1*
OR enter 'Q' to discontinue with this application

Selecting the option shown caused the static analysis program STATIC to be run. At this stage, all the data required for the static analysis of the shaft i.e. the geometry and loadings could be retrieved from the database and hardly any input was required from the user.

The program STATIC uses the finite element method to separately analyze the shaft as a bar, a torsional element and a beam in two planes. The results obtained are summarized below.

STATIC ANALYSIS RESULTS (Bar analysis)

(none as there is no axial loading on the shaft)

STATIC ANALYSIS RESULTS (Torsional analysis)

Node No.	Deflection (mm)	Stress (N/mm2)	Strain
1	0.0	-20	-1.48×10^{-2}
2	-8.52×10^{-4}	-20.12	-1.4×10^{-2}
3	-1.67×10^{-2}	-11.17	-7.78×10^{-3}
4	-1.8×10^{-2}	-3.52	-1.6×10^{-3}
5	-1.85×10^{-2}	-4.62	-1.6×10^{-3}
6	-2.19×10^{-2}	5.17×10^{-6}	2.35×10^{-9}
7	-2.19×10^{-2}	-5.34×10^{-6}	-2.77×10^{-9}
8	-2.19×10^{-2}	-1.99×10^{-6}	-1.38×10^{-9}
9	-2.19×10^{-2}	0	0

STATIC ANALYSIS RESULTS (Bending, vertical plane)

Node	Deflection (mm)	Slope	Moment (Nmm)	Force (N)	Stress (N/mm2)
1	1.85×10^{-4}	-1.61×10^{-5}	1.3	0.0	0.0
2	1.69×10^{-4}	-1.61×10^{-5}	-1.5	-0.4	-0.01
3	-1.57×10^{-4}	-1.26×10^{-5}	-13821	-1316	-2.1
4	-2.75×10^{-4}	-1.08×10^{-5}	-26977	-1315	-1.29

5	-3.78×10^{-4}	-9.81×10^{-6}	-40135	-1315	-1.92
6	-2.65×10^{-4}	1.34×10^{-5}	-28026	1805	-2.31
7	-1.93×10^{-4}	1.52×10^{-5}	-19002	1804	-3.19
8	2.1×10^{-4}	2.0×10^{-5}	0.2	0.9	0.0
9	2.31×10^{-4}	2.0×10^{-5}	-1.4	-2.5	-0.01

STATIC ANALYSIS RESULTS (Bending, horizontal plane)

Node	Deflection (mm)	Slope	Moment (Nmm)	Force (N)	Stress (N/mm2)
1	-6.75×10^{-5}	-1.61×10^{-6}	0.5	-0.3	0.0
2	-6.16×10^{-5}	-1.61×10^{-6}	-0.3	-0.1	-0.01
3	5.71×10^{-5}	-1.26×10^{-6}	-5030.5	478.8	0.76
4	1.0×10^{-4}	-1.08×10^{-6}	-9819	478.8	0.46
5	1.37×10^{-4}	-9.81×10^{-6}	-14608	478.8	0.69
6	9.67×10^{-5}	1.34×10^{-6}	-10201	-657	0.83
7	7.05×10^{-5}	1.52×10^{-6}	-6916	-656.9	1.16
8	-7.67×10^{-5}	2.0×10^{-6}	0.5	-1.3	0.0
9	-8.41×10^{-5}	2.0×10^{-6}	-0.5	-0.6	-0.01

After saving the analysis results to the database, the menu shown below was then displayed enabling the user to quit the static analysis application.

INDUSTRIAL GEARBOX DESIGN

Application level

Job Number EMP7

Configuration: No.1

Date started: 23/02/87

Application: **Shaft Analysis**

Function module	State
Design analysis of shaft No.1	**Completed**
Design analysis of shaft No.2	

Enter the required option (1, ,) *Q*

OR enter 'Q' to discontinue with this application

(e) **Shaft material specification**

Having chosen to leave the static analysis application as shown above, the job menu shown below was then presented.

INDUSTRIAL GEARBOX DESIGN

Job level

Job Number EMP7

Configuration: No.1
Date started: 23/02/87

Job module	State
Gear pair design	**Completed**
Draw shaft	**Incomplete**
Bearing Selection	**Incomplete**
Shaft analysis	**Incomplete**
Material specification	

Enter the required option (G,D,B,S,M) *M*
OR enter V=view configuration, Q=quit this job

Choosing to proceed to shaft material specification caused the display of the corresponding function menu shown below.

INDUSTRIAL GEARBOX DESIGN

Application level

Job Number EMP7

Configuration: No.1
Date started: 23/02/87

Application: **Shaft material specification**

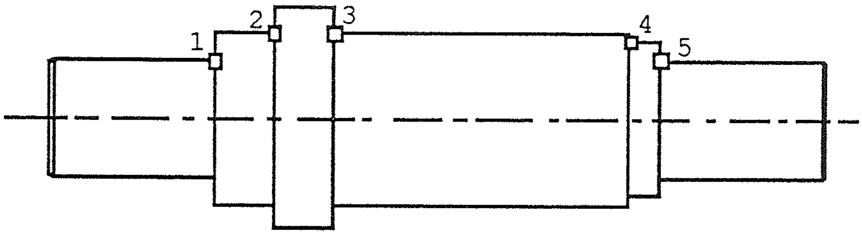
Function module	State
Specify material for shaft No.1	
Specify material for shaft No.2	

Enter the required option (1, ,) *1*
OR enter 'Q' to discontinue with this application

After choosing the option to carry out a material specification as shown above, the program MSELECT was activated: the results summarized below.

BENDING STRESS CONCENTRATION FACTORS

The program first calculated and displayed the theoretical stress concentration factors for bending at the shoulders of the shaft as shown below.



Fillet No.	1	2	3	4	5
Major dia (mm)	53.6	70.2	70.2	53.6	47.0
Minor dia (mm)	35.0	53.6	53.6	47.0	35.0
Fillet rad (mm)	0.25	0.5	0.5	0.5	0.25
Bend. S.C factor	4.323	3.8	3.8	3.109	4.179

The critical section was thereafter determined to be at fillet No.5 where the stresses and fatigue strength reduction factors were then displayed as follows:-

CRITICAL NODE PARAMETERS

Node No.	5
Tensile stress (N/mm^2)	0
Torsional stress (N/mm^2)	-4.168
Vert. plane bend. stress (N/mm^2)	-1.92
Horz. plane bend. stress (N/mm^2)	-0.69
Bend. stress Conc. factor K_t	3.8
Surface factor K_a	0.73
Size factor K_b	0.75
Reliability factor K_c	1
Temperature factor K_d	1
Misc. effects factor K_e	1

MATERIAL SPECIFICATION EQUATION

Having determined the location of the critical node, the values of the constants A_1, A_2 and A_3 in the expression

$$N = \sqrt{\left(\frac{A_1}{S_y} + \frac{A_2}{S_u}\right)^2 + 3\left(\frac{A_3}{S_y}\right)^2}$$

for the safety factor at the critical node (see Appendix 2) were then determined to be as follows:-

$A_1 = 0$ $A_2 = 45.72\ N/mm^2$ $A_3 = 4\ N/mm^2$

After accepting the material specification data obtained and having these saved on the database, the menu shown below was then displayed.

INDUSTRIAL GEARBOX DESIGN

Application level

Job Number EMP7 Configuration: No.1

Date started: 23/02/87

Application: **Shaft material specification**

Function module	State
Specify material for shaft No.1	**Completed**
Specify material for shaft No.2	

Enter the required option (1, ,) *Q*

OR enter 'Q' to discontinue with this application

Choosing to leave the material specification function as shown above caused the job menu shown overleaf to be presented.

INDUSTRIAL GEARBOX DESIGN

Job level

Job Number EMP7

Configuration: No.1
Date started: 23/02/87

Job module	State
Gear pair design	**Completed**
Draw shaft	**Incomplete**
Bearing Selection	**Incomplete**
Shaft analysis	**Incomplete**
Material specification	**Incomplete**

Enter the required option (G,D,B,S,M) *Q*
OR enter V=view configuration, Q=quit this job

Choosing to quit working on this job as shown marked the end of the design of the gear pair and shaft No.1 for Job No. EMP7. The design of the second shaft in the gearbox followed the same sequence of functions as that just illustrated for th first one. It is therefore not separately presented in this appendix.

APPENDIX FOUR

THE DESIGN AND IMPLEMENTATION OF A DATA DICTIONARY/ DIRECTORY SYSTEM

A4.1 Introduction

A data directory/dictionary system (DD/D) has been defined in Chapter 5 as being a depository of the metadata i.e.'data about the data' in a processing system. As its designation implies, the basic function of a DD/D system is to store the location as well as the structure and semantics of the various units of data in system. In a database environment, the existence of a DD/D system greatly enhances the design and maintenance of the database and the processing of requests for information.

The design and implementation of a DD/D system for the integrated CAE package for gearbox design created in the course of the research work is outlined in this appendix. In the research, the system was required to satisfy two main design specifications as follows:-

(a) Exercise control over modifications to existing data structures and definitions. Apart from aiding the design of the initial database structure, this facility is generally required in every elaborate database system for purposes of maintaining data integrity and non-redundancy.

(b) Provide the ability to expose and enable the analysis of the data resident on the database at any time. The ability to examine the existing data structures and definitions frequently and easily was particularly important in the design stages of the CAE database system.

The design was based on a 'self-defining' data model proposed by **Fox, Goti, Miller, Sagawa** [**62**] and is considered applicable to any database system based on the relational data model. The modelling of the system is outlined firstly, followed by a description of its implementation using relational modelling and the dBASE II database management system. The main benefits realized from the use of the DD/D system are then presented in the last section of the appendix.

A4.2 **Design of an Entity-Relationship model to hold conceptual view data**

In the database field, a useful way developed for the description of information in an area of discourse is to classify it into three views as follows [**33,35,62**]:-

(a) The conceptual view which describes the information of interest regardless of how it is to be stored using a particular data model or used by applications,

(b) The internal view which describes how the data elements defined in the conceptual view are stored, including the methods of access, and

(c) The external view which is the view of the data as perceived by individual application programmers.

This classification has been discussed at greater length in Chapter 5 (see Fig 5.2). In the same chapter, Entity-Relationship (E-R) modelling has been shown to be a very useful technique in the design of the conceptual view (see Fig 5.6). This is a top-down approach where the data in the system is described using the following basic units:-

(a) Entities i.e. items or concepts that are of interest,

(b) Relationships i.e. associations of interest between two entities, between an entity and another relationship or between two relationships.

(c) Attributes i.e. those properties of an entity or relationship in its own right.

The three-view approach to the description of information and Entity-Relationship modelling were adopted in the design of a conceptual view for the metadata in the CAE database system developed for gearbox design.

A4.2.1 Entity-Relationship modelling of conceptual view data

An E-R diagram developed for the CAE database system for gearbox design is shown in Fig 7.11. This constituted the universe of discourse in the design of the metadata conceptual model. The development of the E-R model for the metadata had three main steps as follows:-

(1) **Identification of entities.**

The main items in an E-R model such as shown in Fig 7.11 are entities, relationships the attributes. Accordingly, three entities were defined as follows:-

- ENTITYMOD (entity model) whose attributes were to describe each entity defined in the conceptual view for the CAE database.
- FIELDMOD (field model) whose attributes were to describe each field defined in the conceptual view for the CAE database. -
- RELATMOD (relationship model) whose attributes were to describe each relationship defined in the conceptual view for the CAE database.

(2) **Identification of useful relationships.**

In the conceptual view for the CAE database, each defined entity has a number of attributes.

Furthermore, some of these attributes form the identifier-attribute set for the entity. Two relationships were therefore defined between the entities ENTITYMOD and FIELDMOD as follows:-

- KEYFIELD, to associate an entity with its key-attribute set and
- DATAFIELD, to associate an entity with its non-key attributes.

In an E-R model, each relationship defined has a source and a target. Furthermore, the source or target can be either an entity or another relationship. Four relationships of interest were therefore identified, associated with the entity RELATMOD. These are as follows:-

- SRCENT (source entity), to associate a relationship with its source, where the source is an entity,
- TRGENT (target entity), to associate a relationship with its target, where the target is an entity,
- SRCREL (source relationship), to associate a relationship with its source, where the source is a relationship,
- SRCREL (source RELATIONSHIP), to associate a relationship with its target, where the

source is a relationship.

(3) **Allocation of attributes**

Attributes were then allocated to each of the entities ENTITYMOD, FIELDMOD, and RELATMOD. These are detailed in Sect A4.3.1.

The resulting E-R model is shown in Fig A4.1. more entities, relationships and attributes could have been defined but the conceptual view shown was considered to be capable of modelling adequate data on which to base a basic DD/D system.

Note that apart from holding data about the CAE database, the E-R model developed is also capable of holding data about itself i.e. data about the entities ENTITYMOD, FIELDMOD and RELATMOD is held in ENTITYMOD and descriptions of their fields are held in FIELDMOD. Also, data about the six relationships KEYFIELD, DATAFIELD etc., is held in RELATMOD. This is why the model is designated as 'self-defining' in reference [62]. This feature is very useful as it would allow both the main and the metadata conceptual models to be expanded as necessary without the need for separate documentation.

A4.3 Implementation and usage of the data directory/ dictionary system.

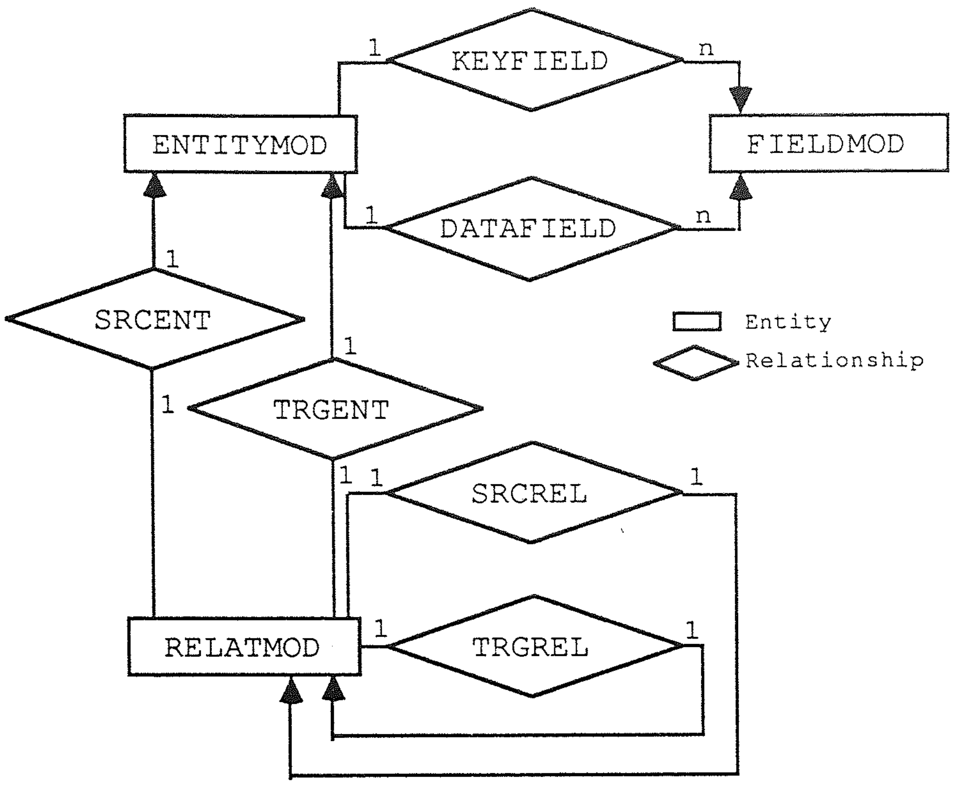


Fig A4.1 Showing an Entity-Relationship model to hold conceptual view information

The metadata conceptual view shown in Fig A4.1 was translated into a schema and implemented using the dBASE II relational database management system. Programs were then written that drew on the data in the schema to satisfy prescribed DD/D requirements. The implementation and usage of the DD/D system is outlined in this section.

A4.3.1 **Attributes allocated to entities**

The attributes allocated to each of the entities and relationships are detailed in the relation schemes shown below. The attributes constituting the identifier-attribute set for the relation are shown in italics. Note that the identifier-attribute set for a relationship is a union of those for the entities and other relationships involved in it.

ENTITYMOD (*ENTITY NAME*, Entity definition)

FIELDMOD (*FIELD NAME*, Field type, Field length, Decimal places, Field definition)

RELATMOD (*RELATIONSHIP NAME*, Relationship type, Relationship source, Relationship target, Relationship function)

KEYFIELD (*ENTITY NAME, FIELD NAME*)

DATAFIELD (*ENTITY NAME, FIELD NAME*)

SRCENT (*RELATIONSHIP NAME, ENTITY NAME*)

TRGENT (*RELATIONSHIP NAME, ENTITY NAME*)

SRCREL (*RELATIONSHIP NAME, RELATIONSHIP NAME*)

TRGENT (*RELATIONSHIP NAME, RELATIONSHIP NAME*)

The relations listed were then implemented as dBASE II data files (see Appendix 1 for a general description of dBASE II relational database management system). These files sit side by side with the main ones for the CAE system for gearbox design and can be manipulated in exactly the same way.

A4.3.2 **The data dictionary/directory program**

After implementing the metadata schema, a program was developed (coded in dBASE II's own programming language) that utilizes the data in the schema to satisfy a variety of queries regarding the data in the CAE system's database, as well as control the modification to the database structure. the program (named QUERY) is menu-driven and, because dBASE II programming allows extensive use of subprograms, each menu can be modified and new menus can be readily added.

The tree structure for the program and the menu details are depicted in Fig A4.2. The menu items marked with asterisks were implemented primarily to aid database design and implementation. These are therefore not available to the general user of the CAE system as they involve modification of the existing database file structures: in any database system these should generally only be carried out by the database designer/adminstrator.

2.4 **The value of the DD/D system to the research work**

The DD/D system described was implemented before the design of the database structure for the CAE system for gearbox design. The main benefits provided by the system were as follows:-

(a) **Reduction of manual documentation.**

The use of the system greatly reduced manual documentation during the design of the main database.

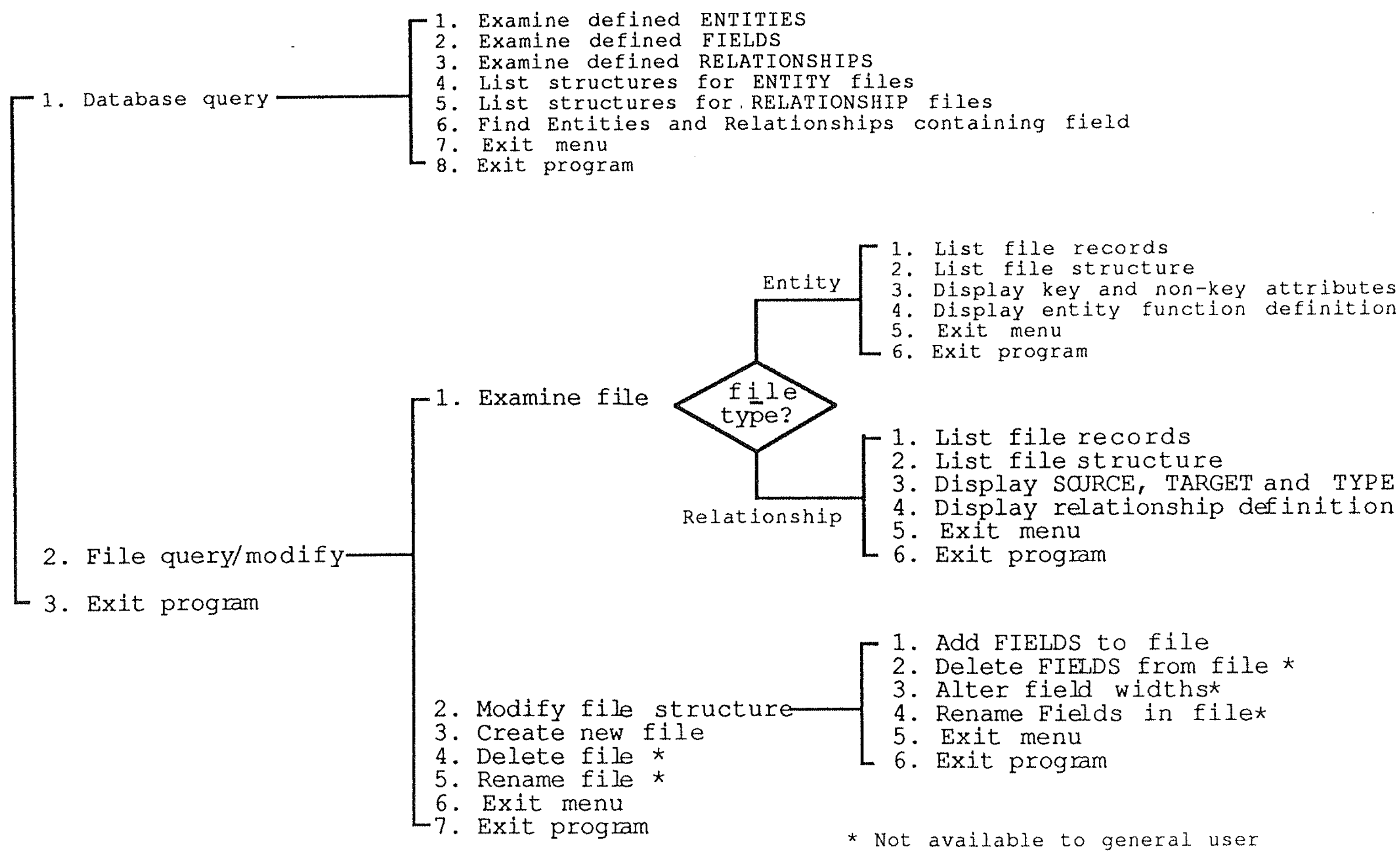


Fig A4.2 Showing a tree structure for the menu-driven data dictionary/directory program

This allowed the design effort to be concentrated on the more important data analysis and modelling issues.

(b) **Provision of quick and accurate displays of existing file structures.**

This proved to be a very useful feature during database design and should be equally beneficial to any future work on the CAE package.

(c) **Exercise of control over modifications to the main database.**

The system maintains the uniqueness of file and field names. This is important for minimizing data redundancy and eliminating errors that can be caused by inconsistent data definition.

A number of possible improvements to the DD/D system can be cited. Apart from expanding the menu sizes and numbers, it would be desirable to implement a facility which would allow limited modification of the database file structures while ensuring that the operations of all the existing programs are not adversely affected. This would increase the flexibility of the whole CAE system and reduce the role of a database adminstrator in the maintenance of the system.

REFERENCES

[1] Hartland-Swann J., 'CIM: Hit or Miss ?'. Industrial Computing, Feb 1986

[2] Institution of Production Engineers, A Management Guide from Concept to Production. (London, 1974)

[3] Mederith D D, Wong K W, Woodhead R W, Wortman R H., Design and Planning of Engineering Systems. (Prentice-Hall Inc., Englewood Cliff, New York, 1973)

[4] Eberlein W. & Wedekind H 'Design Databases in Integrated Engineering Systems', File Structures and Databases for CAD. Proceedings of the IFIP WG5.2 Working Conference on File Structures and Databases for CAD, Seeheim, F R G, 1981. Edited by Encarnacao J. (North-Holland Publishing Co. Amsterdam, 1982)

[5] Hubka V., Principles of Engineering Design. Translated and edited by Eder W E. (Butterworth Scientific, London 1980)

[6] Woodson T T., Introduction to Engineering Design. (McGraw-Hill Book Co., London 1966)

[7] Matousek R., Engineering Design: a systematic approach Edited by Johnson D C. (Blackie & Sons Ltd., London, 1962)

[8] Dixon J R., Design Engineering: Inventiveness, Analysis and Decision-making. (McGraw-Hill Book Co., London, 1966)

[9] Jones J C., Design Methods - seeds of human futures. (John Wiley & Sons Ltd., London 1970)

[10] Pahl G & Beitz W., Engineering Design. (The design Council, London, 1984)

[11] Doyle L E., Manufacturing Processes and Materials for Engineers. (Prentice-Hall Inc., Englewood Cliff, N J, 1961)

[12] Johnson G E & Eary D F., Process Engineering for Manufacturing. (Prentice-Hall Inc., Englewood Cliff NJ, 1962)

[13] Bolz R W., Production Processes. (Penton Publishing Co., London 1963)

[14] Kazanas H C, Baker E G and Gregor T., Basic Manufacturing Processes. (McGraw-Hill Book Co., London 1981)

[15] Cook N H., Manufacturing Analysis. (Addison-Wesley Publishing Co., Reading, Massachusetts, 1966)

[16] Schey J A, Introduction to Manufacturing Processes. (McGraw-Hill Book Co., London, 1977)

[17] Carson J W & Rickards T., Industrial New Product Development - a manual for the 80s. (Gower Press, Farnborough, England, 1979)

[18] Douglas G, Kemp P and Cook J., Systematic New Product Development. (Gower Publishing Co. Ltd., Aldershot, England, 1978)

[19] Crawford C M., New Product Management. (Richard D Irwin Inc., Homewood, Illinois, 1983)

[20] Berridge A E., Product Innovation and Development. (Business Books Ltd., London, 1977)

[21] Kraushar P M., New Products and Diversification. (Business Books Ltd., London, 1969)

[22] Richards T H., 'Producing stress and vibration analysis data for Engineering Design', Engineering, Research and Design - bridging the gap. (Mechanical Engineering Publications Ltd, London, 1981)

[23] Lillihagen F M and Dokken T., 'Towards a methodology for constructing product modelling databases in CAD', File Structures and Databases for CAD. Proceedings of the IFIP WG5.2 Working Conference on File Structures and Databases for CAD, Seeheim, F R G, 1981. Edited by Encarnacao J. (North-Holland Publishing Co. Amsterdam, 1982)

[24] Besant C B., Computer Aided Design and Manufacture. (Ellis-Horwood Ltd., Chichester, 1983)

[25] Kellock B., 'Integrating all aspects of manufacture'. Machinery and Production Engineering, Vol 1, No 15, Dec 1982

[26] Jurocic D and Barr R E., 'Graphics and CAD - a

systematic approach'. Mechanical Engineering, Vol 48, Sept 1982

[27] Institution of Production Engineers. A Guide to CADCAM. (London, 1983)

[28] Ingersoll Engineers. Integrated Manufacture. (IFS Publications Ltd., U.K, 1985)

[29] Looney M, 'Integration - why not ?'. CADCAM International, May 1984

[30] Initial Graphics Exchange Specification Version 1.0. (National Bureau of Standards Publication NBSR 80-1978 (R) 1980)

[31] Sadowski R P., 'The history of Computer Integrated Manufacturing'. Industrial Engineering, Vol 16, No 3, Mar 1984.

[32] French R L, 'Management looking at CIM must deal effectively with these Issues and Realities'. Industrial Engineering, Vol 16, No 8, Aug 1984

[33] Sadowski R P, 'Computer Integrated manufacturing Series will apply systems approach to factory of the future'. Industrial Engineering, Vol 16, No 1, Jan 1984

[34] Jones J. L., 'Report of the CODASYL Data Description Language Committee'. Information Systems, Vol 3, 1978

[35] Tsichritzis D. and Klug A., 'The ANSI/X3/SPARC DBMS FRAMEWORK, Report of the Study Group on Database Management Systems'. Information Systems, Vol 3, 1978

[36] Codd E.F., 'Relational Database: A Practical Foundation for Productivity', Communications of the ACM 25, Feb 1982.

[37] Teorey T J and Fry J P., Design of Database Structures. (Prentice-Hall Inc., Englewood Cliffs, N.J., 1982)

[38] Chaudhuri J., 'Conceptual Modelling', University of Aston Computer Centre, Technical Report TR82006, 1982

[39] Avison D.E., 'Techniques of Data Analysis', University of Aston Computer Centre, Technical Report TR80006, 1980

[40] CODASYL 'Data Base Task Group (DBTG) report', Association for Computing Machinery, April 1971.

[41] Codd E.F., 'A Relational Model of Data for Large Shared Data Banks', Communications of the ACM, vol 13, No.6, June, 1971

[42] Cardenas A.F., Database Management Systems. (Allyn & Bacon Inc., London, 1985)

[43] MacFadden F R and Hoffer J A., Database Management. (Benjamin/Cummings Publishing Co., London, 1985)

[44] Erbelein W and Wedekind H., 'A methodology for embedding design databases into integrated engineering systems', File Structures and Databases for CAD. Proceedings of the IFIP WG5.2 Working Conference on File Structures and Databases for CAD, Seeheim, F R G, 1981. Edited by Encarnacao J. (North-Holland Publishing Co. Amsterdam, 1982)

[45] Lillihagen F M and Dokken T., 'Towards a methodology for contructing product modelling databases in CAD', File Structures and Databases for CAD. Proceedings of the IFIP WG5.2 Working Conference on File Structures and Databases for CAD, Seeheim, F R G, 1981. Edited by Encarnacao J. (North-Holland Publishing Co. Amsterdam, 1982)

[46] Kimura F, Yamaguchi Y, Susaki Y, Kido K and Hosaka M., 'Construction and uses of an engineering database in design and manufacturing environments', File Structures and Databases for CAD. Proceedings of the IFIP WG5.2 Working Conference on File Structures and Databases for CAD, Seeheim, F R G, 1981. Edited by Encarnacao J. (North-Holland Publishing Co. Amsterdam, 1982)

[47] Lorie R A., 'Issues in databases for design applications', File Structures and Databases for CAD. Proceedings of the IFIP WG5.2 Working Conference on File Structures and Databases for CAD, Seeheim, F R G, 1981. Edited by Encarnacao J. (North-Holland Publishing Co. Amsterdam, 1982)

[48] Managaki M., 'A multi-layered database architecture for CADCAM systems', File Structures and Databases for CAD. Proceedings of the IFIP WG5.2 Working Conference on File Structures and Databases for CAD, Seeheim, F R G,

1981. Edited by Encarnacao J. (North-Holland Publishing Co. Amsterdam, 1982)

[49] Katz R H., Information Management for Engineering design. (Springer-Verlag, Berlin, 1985)

[50] Katz R H., 'Transaction Management in the design environment'. New applications of databases. Edited by Gardarin G and Gelembe E. (Academic Press (London) Ltd, London, 1984)

[51] Gray M., 'Databases for computer aided design'. New applications of databases. Edited by Gardarin G and Gelembe E. (Academic Press (London) Ltd, London, 1984)

[52] Grabowski H and Eigner M., 'A data model for a design database', File Structures and Databases for CAD. Proceedings of the IFIP WG5.2 Working Conference on File Structures and Databases for CAD, Seeheim, F R G, 1981. Edited by Encarnacao J. (North-Holland Publishing Co. Amsterdam, 1982)

[53] Baron N, Bornkessel E, Cullman N, Klos W F and Magalhaes L P., 'An approach to the integration of geometric capabilities into a database for CAD applications', File Structures and Databases for CAD. Proceedings of the IFIP WG5.2 Working Conference on File Structures and Databases for CAD, Seeheim, F R G, 1981. Edited by Encarnacao J. (North-Holland Publishing Co. Amsterdam, 1982)

[54] Heerema F J and Kreijkamp H A., 'A multi-discipline and integrated facility for management and evaluation of engineering data', CAD84 - 6th International conference and exhibition on computers in engineering, Brighton, 1984. Edited by Wexler J. (Butterworth & Co. (Publishers) Ltd., Guildford, 1984)

[55] Steenbergen H and Heerema F J., 'Database Adminstrator facilities for engineering data management'. Engineering Software IV - proceedings of the 4th International conference, London, 1985. Edited by Adey R A. (Springer-Verlag, Berlin, 1985)

[56] Lawrence A., 'CIM architecture', Industrial Computing, Sept 1986.

[57] Jacke M and Vassiliou Y., 'Databases and expert systems: opportunities and architectures for integration', New Applications of Databases, edited by Gardarin G and Gelembe E. (Academic Press, London, 1984)

[58] Peterson R E., Stress Concentration Factors: Charts and relations useful in making strength calculations for machine parts and structural elements. (John Wiley & Sons, London, 1974)

[59] Remenyi D., dBASE II and dBASE III. (Longmans Publishing Co., Harlow, 1985)

[60] Shaw T., dBASE II: Developing Applications. (Addison-Wesley Publishing Co., London 1984)

[61] Burns G., Understanding dBASE III and dBASE II. (Sigma Press, Cheshire, 1985)

[62] Fox J M, Goti J C, Miller C R and Sagawa J M., 'Implementing a self-defining Entity-Relationship model to hold conceptual view information'. Entity-Relationship Approach to Information modelling and analysis. Proceedings of the 2nd International conference on Entity-Relationship approach, Washington DC, 1981. Edited by Chen P P. (North-Holland Publishing Co., Oxford, 1983)

[63] Shigley J., Mechanical Engineering Design. (McGraw-Hill Book Co., London 1985)

[64] Martin J., Computer data-base organization. (Prentice-Hall Inc., Englewood Cliffs, N.J., 1975)