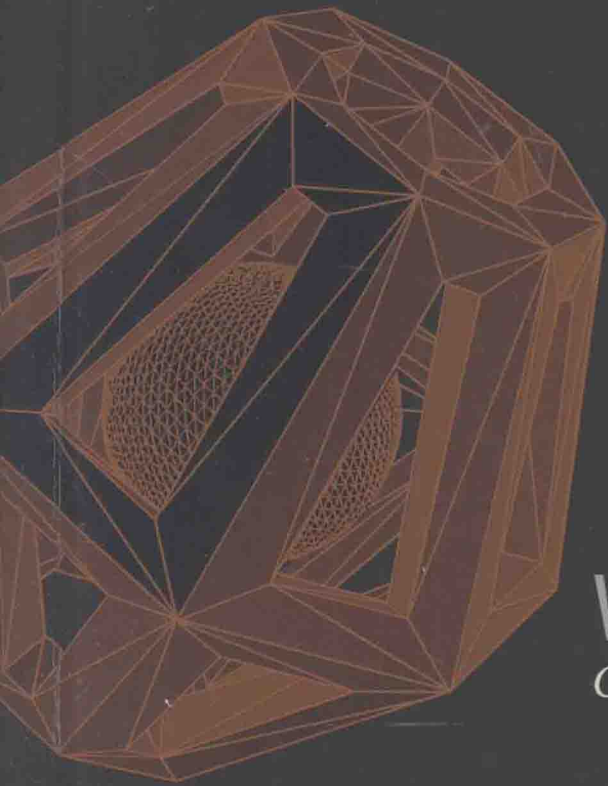


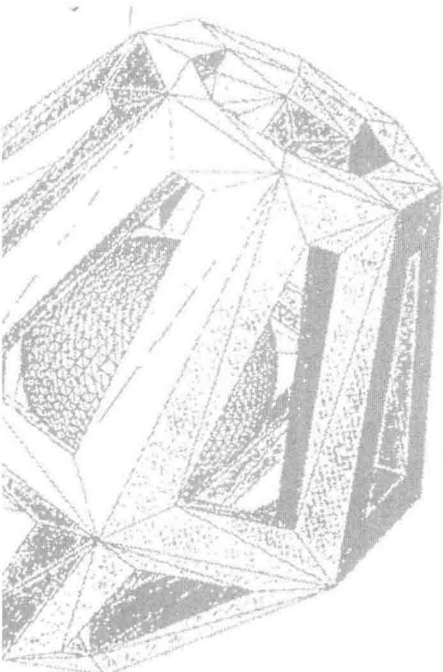
Vol. 1

Computer Techniques

Computer Aided and Integrated Manufacturing System

A 5-Volume Set

Cornelius T Leondes



Vol. 1
Computer Techniques

Computer Aided and Integrated Manufacturing Systems

A 5-Volume Set

Cornelius T Leondes
University of California, Los Angeles, USA

World Scientific
New Jersey • London • Singapore • Hong Kong

Published by

World Scientific Publishing Co. Pte. Ltd.
5 Toh Tuck Link, Singapore 596224
USA office: Suite 202, 1060 Main Street, River Edge, NJ 07661
UK office: 57 Shelton Street, Covent Garden, London WC2H 9HE

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

COMPUTER AIDED AND INTEGRATED MANUFACTURING SYSTEMS
A 5-Volume Set
Volume 1: Computer Techniques

ISBN 981-238-339-5 (Set)
ISBN 981-238-983-0 (Vol. 1)

Typeset by Stallion Press

Printed by Fulsland Offset Printing (S) Pte Ltd, Singapore

Preface

Computer Technology

This 5 volume MRW (Major Reference Work) is entitled "Computer Aided and Integrated Manufacturing Systems". A brief summary description of each of the 5 volumes will be noted in their respective PREFACES. An MRW is normally on a broad subject of major importance on the international scene. Because of the breadth of a major subject area, an MRW will normally consist of an integrated set of distinctly titled and well-integrated volumes each of which occupies a major role in the broad subject of the MRW. MRWs are normally required when a given major subject cannot be adequately treated in a single volume or, for that matter, by a single author or coauthors.

Normally, the individual chapter authors for the respective volumes of an MRW will be among the leading contributors on the international scene in the subject area of their chapter. The great breadth and significance of the subject of this MRW evidently calls for treatment by means of an MRW.

As will be noted later in this preface, the technology and techniques utilized in the methods of computer aided and integrated manufacturing systems have produced and will, no doubt, continue to produce significant annual improvement in productivity — the goods and services produced from each hour of work. In addition, as will be noted later in this preface, the positive economic implications of constant annual improvements in productivity have very positive implications for national economies as, in fact, might be expected.

Before getting into these matters, it is perhaps interesting to briefly touch on Moore's Law for integrated circuits because, while Moore's Law is in an entirely different area, some significant and somewhat interesting parallels can be seen. In 1965, Gordon Moore, cofounder of INTEL made the observation that the number of transistors per square inch on integrated circuits could be expected to double every year for the foreseeable future. In subsequent years, the pace slowed down a bit, but density has doubled approximately every 18 months, and this is the current definition of Moore's Law. Currently, experts, including Moore himself, expect Moore's Law to hold for at least another decade and a half. This is hugely impressive with many significant implications in technology and economics on the international scene. With these observations in mind, we now turn our attention to the greatly significant and broad subject area of this MRW.

"The Magic Elixir of Productivity" is the title of a significant editorial which appeared in the *Wall Street Journal.* While the focus in this editorial was on productivity trends in the United States and the significant positive implications for the economy in the United States, the issues addressed apply, in general, to developed economies on the international scene.

Economists split productivity growth into two components: Capital Deepening which refers to expenditures in capital equipment, particularly IT (Information Technology) equipment: and what is called Multifactor Productivity Growth, in which existing resources of capital and labor are utilized more effectively. It is observed by economists that Multifactor Productivity Growth is a better gauge of true productivity. In fact, computer aided and integrated manufacturing systems are, in essence, Multifactor Productivity Growth in the hugely important manufacturing sector of global economics. Finally, in the United States, although there are various estimates by economists on what the annual growth in productivity might be, Chairman of the Federal Reserve Board, Alan Greenspan — the one economist whose opinions actually count, remains an optimist that actual annual productivity gains can be expected to be close to 3% for the next 5 to 10 years. Further, the Treasure Secretary in the President's Cabinet is of the view that the potential for productivity gains in the US economy is higher than we realize. He observes that the penetration of good ideas suggests that we are still at the 20 to 30% level of what is possible.

The economic implications of significant annual growth in productivity are huge. A half-percentage point rise in annual productivity adds $1.2 trillion to the federal budget revenues over a period of 10 years. This means, of course, that an annual growth rate of 2.5 to 3% in productivity over 10 years would generate anywhere from $6 to $7 trillion in federal budget revenues over that time period and, of course, that is hugely significant. Further, the faster productivity rises, the faster wages climb. That is obviously good for workers, but it also means more taxes flowing into social security. This, of course, strengthens the social security program. Further, the annual productivity growth rate is a significant factor in controlling the growth rate of inflation. This continuing annual growth in productivity can be compared with Moore's Law, both with huge implications for the economy.

The respective volumes of this MRW "Computer Aided and Integrated Manufacturing Systems" are entitled:

Volume 1: Computer Techniques
Volume 2: Intelligent Systems Technology
Volume 3: Optimization Methods
Volume 4: Computer Aided Design/Computer Aided Manufacturing (CAD/CAM)
Volume 5: Manufacturing Process

A description of the contents of each of the volumes is included in the PREFACE for that respective volume.

Computer Techniques is the subject for Volume 1. In this volume, computer techniques are shown to have significance in the design phase of products. These techniques also have implications in the rapid prototyping phase of products, automated workpiece classification, reduction or elimination of product errors in manufacturing systems, on-line process quality improvements, etc. These and numerous other topics are treated comprehensively in Volume 1.

As noted earlier, this MRW (Major Reference Work) on "Computer Aided and Integrated Manufacturing Systems" consists of 5 distinctly titled and well-integrated volumes. It is appropriate to mention that each of the volumes can be utilized individually. The significance and the potential pervasiveness of the very broad subject of this MRW certainly suggests the clear requirement of an MRW for a comprehensive treatment. All the contributors to this MRW are to be highly commended for their splendid contributions that will provide a significant and unique reference source for students, research workers, practitioners, computer scientists and others, as well as institutional libraries on the international scene for years to come.

Contents

CHAPTER 1

COMPUTER TECHNIQUES AND APPLICATIONS IN THE CONCEPTUAL DESIGN PHASE OF MECHANICAL PRODUCTS

WYNNE HSU and IRENE M. Y. WOON

School of Computing
National University of Singapore
Lower Kent Ridge Road
Singapore 119260
{whsu,iwoon}@comp.nus.edu.sg

The conceptual stage of the design process is characterized by a high degree of uncertainty concerning the design requirements, information and constraints. However, decisions made at this early stage have a significant influence on factors such as costs, performance, reliability, safety and environmental impact of a product. More importantly, a poorly conceived design can never be compensated for in the later stages of design. There is some controversy over the use of computers at this stage of product design. Some researchers feel that providing accuracy during this phase when solutions are imprecise, ill defined, approximate or unknown, accurate calculations impart a false sense of confidence in the validity of the solution. Others feel that maturing computer techniques with richer representations can provide invaluable assistance in specific sub-tasks of this phase. The purpose of this paper is to review advances in computational support for conceptual design from its early days to its current position. For each technique, we follow its progress from its conception to its latest status, pointing out significant variations and trends.

Keywords: Conceptual design; computer-aided design; mechanical product design; conceptual design models.

1. Introduction

The conceptual stage of the design process is one of the most imaginative stages of the design process in which human creativity, intuition and successful past experience play an important role. This early stage of the design is identified with a high degree of uncertainty concerning the design information and lack of clarity of the design brief (i.e. mission, instructions). The design of mechanical products is complex (as opposed to well-defined domains such as VLSI design) because they are, in general, multi-faceted. Some attributes of the task related to conceptual design process can be summarized as:

(1) Analysis of many dimensions of the problem in search of possible solutions.
(2) Synthesis of a number of possible solutions within a framework of constraints and requirements set forth in the design brief.
(3) Critical evaluation of alternative solutions.
(4) Selection of the design option that best fits the purpose.

These activities are highly non-linear and non-algorithmic by nature. There are no predefined rules for formulating design solutions. At present, design solutions developed mainly rely on heuristics and past experience.

There is some controversy over the use of computers in the early stages of product design. One school of thought is of the opinion that at the early stages of design where solutions are ill defined, accurate calculations impart a false sense of confidence in the validity of the solution. They support the practice of using heuristics that are relatively simple and less accurate than algorithmic techniques. The counter argument is that computers can generate and handle complex representations with ease and so even at the very early stages of the design process, one could introduce advanced algorithmic techniques. Hence, even though the designer may not have determined the design parameters to a high level of accuracy, one has not introduced further inaccuracy through the algorithm. In addition, there is also the concern that the instant we analyze a situation in terms of properties, artifacts, etc. we limit our view of the problem to that which can be expressed in modeling paradigm. For example, in the expert system to select a bridge type, only the structural designs built into the expert system can be designed. This creates 'blindness' for all other kinds of possible designs. The counter argument is that as computer techniques mature, with richer representations, more background knowledge and deep knowledge (or reasoning from first principles), they can perform well in real-world applications. This avoids the problem of blindness creation in the early stages of product design.

The purpose of this paper is to review advances in computational support for conceptual design from its early days to its current position. We define 'early days' as techniques and applications that originated before the 1980s, the 'recent past' as developments that originated from 1980s–1990s, and the 'current scene', as developments that germinated from 1997. For each technique, we follow its progress from its conception to its latest status, pointing out significant variations and trends. The later techniques tend to exhibit a 'hybrid' approach, reflecting the inadequacy of any single technique in supporting this complex phase of product creation.

2. Early Days

The predominant techniques used to support conceptual design in its early days (1960s to 1980s) are systems that were built on languages, images, graphs and operation research techniques. Computer technology, both hardware and software, were immature at this time. In this section, we look at how such systems have matured over the years to support the complex task of conceptual design.

2.1. *Languages*

Language represents an attempt at formalizing design. It is useful in expressing our understanding of designs unambiguously. In general, a language is defined by a grammar. A grammar is denoted by the quintuple (T, N, S, P) where T is the set of terminals, N is the set of non-terminals, S is the start symbol and P is the set of production rules. Table 1[1] gives an example of how part of a grammar (expressed in BNF specification language) can be used to describe the positions and motions of each part of a fixed axes mechanism and their relationship between them. The terminals are expressed in bold fonts. The non-terminals are expressed in normal fonts. The start symbol is *Motion* and the production rules are listed in Table 1.

Due to its compact representations, grammar/language is an efficient means of structuring design knowledge. Indeed, many pieces of work have used language/grammar as the underlying representation for their design knowledge. For example, Rinderle[2,3] used a graph-based language to describe behavioral specifications of design as well as the behavior of the components. Neville and Joskowicz[1] present a language for describing the behavior of fixed-axes mechanism e.g. couplers, indexers and dwells. Predicates and algebraic relations are used to describe the positions and motions of each part. Vescovi *et al.*[4] developed a language, CFRL, for specifying the causal functionality of engineered devices. In terms of grammar, Carlson,[5] Stiny[6] and Heisserman[7] have looked into using shape and/or spatial grammars to express physical design forms. In particular, Mitchell[8] has combined shape grammars with simulated annealing to tackle the problem of free-form structural design. First, shape grammars are used to generate structural design possibilities. Then, stochastic optimization of all the possible designs are achieved using simulated annealing. This allows the generation of large number of sound, efficient free-form solutions that otherwise would never have been imagined. A number of researchers[9]–[13] have also made use of grammars in engineering applications. Tyugu[14] also proposed an attribute model based on attribute grammar for representing implementation knowledge of design objects. Similarly, Andersson *et al.*[15] proposes a modeling language, CANDLE, which enables the use of engineering terminology to support early design phases of mechanisms and manipulator systems. In CANDLE, the basic taxonomies of engineering terminology are augmented with

Table 1. Example of part of a grammar.

Motion ::= SimpleMotion | ComplexMotion
SimpleMotion ::= <Part, SM_Type, Axis, InitialPosition, Extent, Relations>
SM_Type ::= **Translate** | **Rotate** | **Screw** | **Translateand Rotate** | **Stationary** | **Hold**
Extent ::= AxisParameter **by** Amount
Amount ::= **Real** | **Constant** | **Variable** | **Infinity**

the physical and solution principles that are specific for the design of mechanisms and manipulator systems.

In fact, the general approach adopted by researchers is that they would propose different special purpose languages to describe some aspects of design that they are interested in modeling. This approach of developing special-purpose languages works well for non-collaborative design effort. With the increasing emphasis on the use of product design as a firm's competitive advantage, the trend is towards supporting concurrent collaborative design. Hence, we find that effort is directed to developing shareable design ontology. An ontology is a useful set of terms/concepts that are general enough to describe different types of knowledge in different domains but specific enough to do justice to the particular nature of the task at hand. Alberts[16] proposed YMIR as an engineering design ontology. The "How Things Work" project at Stanford University[4] aims to build a large-scale ontology of engineering knowledge. By having a common set of ontology, knowledge can be reused and shared. This allows better integration between the different phases of the product's life cycle.[17]

2.2. *Graphs*

Graphs and trees are popular representations in the conceptual design stage. They have been used to model all aspects of a product — function, behavior, and structure. Function is the perceived use of the device by the human being. Behavior is the sequence of states in which the device goes through to achieve the function. Structure refers to the physical components or forms that are utilized to achieve the behavior. Kuipers[18] illustrates this distinction with the example of a steam valve in a boiler. The function of the steam valve is to prevent an explosion, its behavior is that it opens when a certain pressure difference is detected and its structure is the physical layout and connection between the various physical components.

Malmqvist[19] demonstrates how graphs can be used to model the functions of structural systems in mechanics, electronics, hydraulics e.g. hole punch, washing machine. Nodes of graphs are lumped elements which correspond to the different physical properties (capacitance, transformers) and these nodes are connected by edges (bonds) e.g. force, velocity. The power flow direction and causality of bonds are specified. Murthy and Addanki[20] manipulate a graph of models to modify a given prototype of some structural engineering system e.g. design of beams. A model describes the behavior of the system under certain explicit assumptions. The models form the nodes in a graph and the edges represent sets of assumptions that must be added or relaxed to go between adjacent models. Graph/trees have also been used to model the physical representations of the design components and their layout.[21,22] Besides modeling structural, behavioral and functional aspects of the product, graph and trees have also been used to model requirements and constraints.[23] Kusiak and Szczerbicki[24] use tree models in the specification stage of conceptual design to represent the functions and requirements of mechanical systems, with an incidence

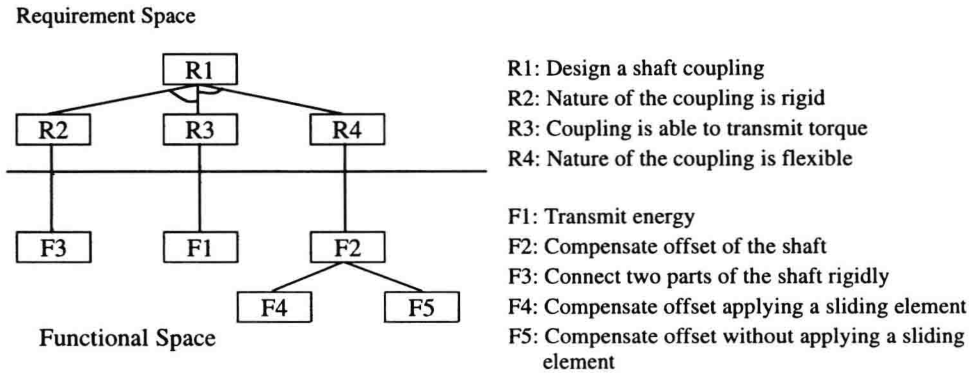


Fig. 1. An example of graph model.

matrix to represent the interaction between requirements and functions. Figure 1 shows the requirement and functional tree for the design of a shaft coupling.

An arc between the nodes of a tree represents a conjunction. A node without an arc represents a disjunction. There are therefore two sets of requirements that satisfy $R1$: $\{R2, R3\}$ and $\{R3, R4\}$.

2.3. *Images*

Perhaps the closest to human's way of thinking and reasoning is through the use of visual thinking models. Visual thinking has its beginning since 1969.[25] It did not gain a high profile in design research until McKim[26] demonstrated through experimental studies that visual thinking is vital to all branches of design practice. Freehand sketching is good for accelerating discussions and for comparing different solutions.[27]

Hand-sketched diagrams are also good in allowing different views of the sketch so as to obtain a good spatial image of the design solution.[28] In 1990, Radcliffe and Lee[29] proposed a model for the process of visual thinking that overcomes the barrier between the cognitive processes and the physical domain. Sittas[30] further explored the issues involved in supporting the creation and manipulation of 3D geometry during the conceptual design sketching activity.

2.4. *Operation Research models*

Operation Research (OR) emphasizes structured, numeric models, where a model is expressed in equations and the design goal, as one or many objective functions e.g. minimum weight, size or cost. Systems built with such underlying models endeavor to find values of variables that meet the equations and maximize/minimize one or several objective functions. In general, design problems are represented as follows. Let the continuous variables be x and the discrete variables be y.

The parameters which are normally specified as fixed values are represented by theta (θ). The design goal (or goals) can be expressed as the objective function $F(x, y, \theta)$. This function is a scalar for a single criterion optimization, and a vector of functions for a multi-objective optimization. Equations and inequality constraints can be represented as vectors of functions, h and g, that must satisfy,

$$h(x, y, \theta) = 0$$
$$g(x, y, \theta) \leq 0 .$$

Many techniques have been proposed to solve optimization problem. A survey of the state-of-the-art optimization techniques in structural design can be found in Koski.[31] The focus of the survey is primarily based on the Pareto optimality concept. Briefly, Koski classified the multi-criteria structural design process into three phases. The first phase is the problem formulation where the criteria, constraints and design variables are chosen. The second phase is the generation of Pareto optimal solutions. The final phase describes the decision-making procedure employed to select the best compromise solution. In another paper by Levary,[32] he draws attention to the interaction between operation research techniques and engineering design. Specific applications of operation research methods are discussed with respect to the following engineering disciplines: computer engineering, communication system engineering, aerospace engineering, chemical engineering, structural engineering and electrical engineering.

A major advantage of OR models is that they provide a great deal of explanatory power in applications where they do apply. However, they are not always easy to apply because the data required by the algorithms may not be available, their scope of applicability is narrow and the algorithms used may not be able to provide optimal solutions because of the problem's complexity.

3. Recent Past

Conceptual design is an engineering activity that is generally ill-structured as it is performed early in a product life cycle, where complete and exact information and knowledge of requirements, constraints etc. is difficult to obtain. This highly skilled task is very complex and requires a mixture not only of different sources of knowledge (e.g. costing, performance, environmental issues) but also different types of knowledge (e.g. physical, mathematical, experiential).[33] The need to integrate different sources and types of knowledge is the emphasis of artificial intelligence research which gained prominence in the early 1980s and sparked off development in the areas such as object-oriented modeling, geometric modeling, case-based modeling and knowledge-based modeling. For example, object-oriented modeling has its roots in frames, an established knowledge representation scheme. Each of these areas has made significant impact on the conceptual design process, as we will see in the following subsections.

3.1. *Geometry models*

Geometry models focus on representing the structural aspects of a product. The objective is to represent 2-dimensional or 3-dimensional geometric shapes in a computer.[34] Popular representations of geometric shapes include: B-rep (boundary representation), CSG (constructive solid geometry), variational geometry and feature representations.

B-rep represents geometry in terms of its boundaries and topological relations. The transformation from one topology to another can be achieved using Euler operators. Since Euler operators are sound,[35] the topological validity of the structure is guaranteed. The major limitation of B-rep is its inefficiency in performing geometric reasoning. While in a B-rep approach, a shape is represented by the boundary information such as faces, edges and vertices, the CSG approach models geometric shapes using a set of primitives such as a cube, cylinder or a prism. Complex shapes are built from the primitives through a set of operators (union, difference and intersection). For example, the primitives given in Fig. 2 can be combined using set operations to form complex solids like that given in Fig. 3.

Although CSG is a geometry modeling technique that was widely accepted by both the research community and industry, it faces several inherent limitations. The most serious limitation, in our opinion, is the non-uniqueness of the CSG representations. This non-uniqueness of representations makes recognition of shapes from

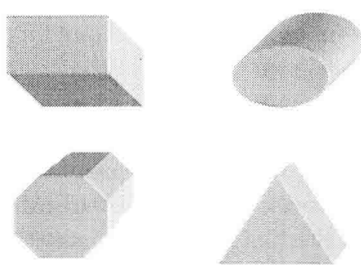


Fig. 2. Some CSG primitives.

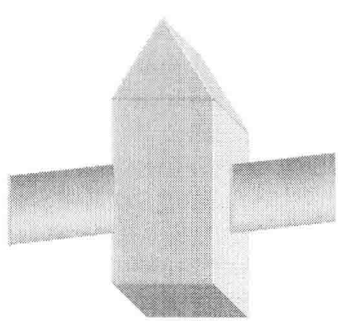


Fig. 3. Complex solid example.

CSG representation extremely difficult. Hence, this tends to dissuade researchers from relying solely on CSG representations alone. In addition, CSG representation does not guarantee that the solid it models is always a valid object. It is possible in CSG representation to model an invalid solid.

Variational modeling allows a designer to use equations to model mechanical components analytically and is popular because it allows the evaluation of competing alternatives. The concept of using variational geometry in computer aided design started as early as 1981. Lin[36] in his thesis described the feasibility of using variational geometry to model geometric information. Light and Gossard[37] expanded upon his work to allow modification of geometric models through variational geometry. Variational geometry design while general and flexible, necessitates the intensive use of numerical solvers to solve many simultaneous nonlinear equations. Frequently, solvers cannot solve these equations. Shpitalni and Lipson[38] combine parametric design with geometric design to ensure that the resulting system is both flexible and guaranteed to find a solution. This system was tested in the designing of sheet metal parts.

In the feature representation approach,[39,40] a part is built from a set of primitive building blocks with the guarantee that this set of building blocks are manufacturable. The notion of features was first proposed as form features[39,41] to bridge the gap between units of the designer's perception of forms and data in geometric models. Shapes are described as the way the designer understands them. A feature-based design approach allows a user to use mechanical features stored in a feature library in his design.[42–45] It provides a means for building a complete CAD database with mechanical features right from the start of the design. However, this approach suffers from the difficulty of a limited number of available feature primitives. It is difficult to satisfy various design needs and in the event that the features interact with one another, new features may arise that can cause complication in the analysis process. EDISON[46] is an example of a system using feature-based modeling. It has a database of known mechanisms and is indexed by their functions, structures and situations in which they are used. Thus far, the majority of feature-based research focuses on using feature-based design for process planning[47,48] and feature recognition.[49] Han and Requicha[50] proposed a novel feature finder that automatically generates a part interpretation in terms of machining features. The feature finder strives to produce a desirable interpretation of the part as quickly as possible. Alternate interpretations could be generated if the initial interpretation was found to be unacceptable by a process planner.

Recently, the trend has been towards the integration of various representation schemes. Keirouz *et al.*[51] proposed an integration of parametric, geometry, features, and variational modeling. With this integration, they showed that the system is able to handle geometry and "what if" questions arising in conceptual design.

In all the above approaches, the assumption is that the support of surface features is well defined on prismatic objects. This is not the case for sculptured surface models and current methods often lead to data explosion. Elsas and Vergeest[52] proposed a displacement feature modeling approach. In this approach, explicit modeling of protrusions and depressions is done in free-form B-spline surfaces that can achieve real-time response and with unprecedented flexibility.

3.2. *Knowledge-based models*

One major development in the recent past is the introduction of knowledge-based models. Knowledge-based models are used to capture procedural design knowledge as well as product or domain knowledge. A prominent branch of knowledge based models is the production model which uses rule representation to facilitate high-level reasoning. The rule based paradigm is adopted by Rao[53] to give advice on which alternative should be chosen in the design of ball bearings. An example of a rule is given in Fig. 4.

Besides rule representation, frame representation is also widely used. In the paper by Tong and Gomory,[54] he used a frame-based structure to model parts of standard kitchen appliances and light sources.

The underlying reasoning techniques used in production models include abductive, deductive, constraint-based, and non-monotonic reasoning. Abductive reasoning says that:

> The surprising fact C is observed;
> But if A were true, C would be a matter of course.
> Hence there is reason to suspect that A is true.

In other words, abductive reasoning (goal directed) tries to derive the premises of a stated conclusion. On the other hand, deductive reasoning says that:

> Suppose if A is true, then C would be a matter of course,
> Now, we observe the fact A.
> We can conclude that C is true.

Hence, deductive search (data driven) moves to arrive at some conclusion, given the initial facts.

An example of an abductive search strategy is given in Tong and Gromory[54] in the design of small electromechanical appliances. Rao[53] shows the use of deductive search strategy in selecting the appropriate ball bearings' design for a set of input parameters e.g. load type, bearing speed, environment of use, etc. Arpaia *et al.*[55] and Carstoiu *et al.*[56] makes use of both patterns of reasoning, the former in

```
If feature is SLOTA and
        If interactingFeature is SlotB and
              If typeofInteraction is intersecting then
                      Send the message intersectingWIth: SlotB to SlotA
                      To get edge entities of SlotA based on typeof Interaction
```

Fig. 4. An example of rule representation.

the design of measurement systems, in mapping from the logical attributes to the physical components of the instrument and the latter in the design of gears. Typically, abductive and deductive reasoning will face the problem of scaling-up. To address this problem, constraint-based reasoning is introduced. Further elaboration on constraint-based reasoning is given in Subsec. 3.3.

There are at present, a number of tools which couple knowledge based systems with conventional systems. Krause and Schlingheider[33] gives a comprehensive overview of such tools e.g. ICAD, MEDUSA-ENGIN, CONNEX. Increasingly, these tools are addressing the problematic areas of development and design.[57,58] Recent development has been towards the concept of metamodels. A metamodel is a qualitative model of causal relationships among all the concepts used for representing the design object.[59] The metamodel reflects the designer's mental model about the structure and behavior of the design object. Metamodel mechanisms include the primary model (a description of the requirement given by the designer) and aspect models (qualitative and quantitative models focusing on specific aspects of the design object).

Though there have been many successful applications that are built upon knowledge models presented here, a number of issues still remain unresolved. Some of these issues include: the verification of the correctness of knowledge models, the handling of incomplete knowledge, the resolution of inherent contradictions that are present in knowledge models and the incremental addition of new knowledge to existing knowledge models.

3.3. *Constraint-based models*

Constraint-based models rely on the designer's experience to select the bounds of design variables that define the search space in which the constraints are processed. A large search space as is expected in real world applications may have all the feasible solutions but it may contain a large number of infeasible solutions. However, if the search space is too restricted (as when particular variables have to be optimal), the risk is that no feasible solution exists in this small space.

A constraint is a statement about a design, the truthfulness of which does not depend on any tradeoffs with goals. For example, the manufacturing cost of the product of around \$100 is a constraint whereas a manufacturing cost objective is to have the product manufactured at the lowest cost. In many instances, it may be possible to translate an objective into constraints e.g. the objective "minimize manufacturing cost" could be stated as manufacturing cost should be less than or equal to \$100. Harmer *et al.*[60] shows how the functional requirements of a product is written as a set of constraints and translated into a desired property profile (which includes functions and objectives) to be matched against that of the existing components in an engineering catalogue. Kolb and Bailey[61] specify constraints between objects derived from analyzing the design of an aircraft engine, and employ a constraint propagation technique to integrate and perform mathematical analyses