FEATURE PAIR BASED DESIGN: DEFINING AND APPLYING FUNCTIONAL RELATIONSHIPS BETWEEN COMPONENTS IN ASSEMBLIES

By

SUDHEER GADDAM

Bachelor of Engineering

Osmania University

Hyderabad, India

1993

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
May, 1995

FEATURE PAIR BASED DESIGN: DEFINING AND
APPLYING FUNCTIONAL RELATIONSHIPS
BETWEEN COMPONENTS
IN ASSEMBLIES

Thesis Approved:

Thesis Adviser

Dean of the Graduate College

ACKNOWLEDGEMENTS

I take this opportunity to express my deepest gratitude and sincere appreciation to Dr. Delahoussaye for his exceptional guidance, patience, encouragement, inspiration and personal interest throughout the research period. I also thank him for guiding and honing my abilities in CAD concepts and helping me at times of difficulty. I am thankful to Dr.Spitler and Dr. Ghajar for serving in my graduate committee.

I thank my brother, Venkata Rama Reddy for his blessing, consistent encouragement, endless love and support. I thank my parents, G. Ram Reddy and G. Vimala, for supporting during my study. I thank Chamu and friends for their encouragement and support though far away from me. I thank my colleage Atul Deosthali for his help. I thank my friends Krishna, Chandrashekar and Suresh for their encouragement and support.

I take this opportunity to thank Rod McAbee, Enfaa Douglas, Clay Couger, Jeff and Charlie for their coordination and support they extended to me in finishing my work successfully.

TABLE OF CONTENTS

Chapter | Page

I. INTRODUCTION 1

Research Problem 1
Objective 4

II. LITERATURE REVIEW 7

Overview 7
Background of Designing Processes 8
Overview of Features 11
Definition of Features 11
Feature Representation 12
The Feature Geometry Design Database 13
Different Approaches implemented by Feature Based Design Systems 14
Different Feature Based Modelling Systems Implemented 18
Feature Recognition 18
Design by Features 22
Interactive Feature Definition 25
Functional Requirements of Features 28
Assembly of different components 29
Assembly Representation 30
Importance of Dimensions and Tolerances in Assembly 33

III. CATALOGUING OF FEATURE-PAIRS DEPENDING ON FUNCTIONAL RELATIONSHIPS 37

Chapter Page

Overview 37
Representation of Features in the System........ 38
Concept of Feature-pairs 42
Tolerances and Clearances in Assembly 52
Cataloguing of Feature-Pair Depending on Functional Relations 58
Manufacturing Details Depending upon the Functional
Relationships of Feature-Pair 61

IV. IMPLEMENTING THE FUNCTIONAL RELATIONSHIPS
IN THE SYSTEM 64

Overview of the System Implemented........ 64
The User Interface 65
Functioning of the System 73

V. CONCLUSIONS AND RECOMMENDATIONS 77

Conclusions........ 77
Recommendations 78

REFERENCES 79

LIST OF TABLES

Table | Page

2.1 ANSI Y14.5 tolerances 32

LIST OF FIGURES

Figure		Page
1.1	Schematic Diagram for the Feature-Pair Based Design System	5
2.1	Different Models of Design Process	9
2.2	Feature Representation in Feature-Based Design	9
2.3	Different Feature-based Modeling Techniques	16
2.4	Geometric Tolerancing Demonstrated by Alan Fleming	34
3.1	Different components before assembly (block-with-pin and block-with-slot)	45
3.2	Assembly of pin-in-slot	46
3.3	The data structure used to represent the assembly of components	46
3.4	Assembly using feature-pair	47
3.5	The data structure to represent the assembly using feature-pairs	47
3.6	Mechanical System involving slider-crank mechanisms and pin joints	49
3.7	Two components of the assembly which consist of mating feature-pair	50
3.8	The tree structure implemented in the system to represent assemblies	51
3.9	i. The Clearance fit	55
	ii. The Interference fit	55
	iii. The Transition fit	55

Figures Page

3.10 i. When the holes are far apart 57

ii. When the holes are closest 57

iii. True positioning with respect to Datum axes 57

3.11 Showing the different relationships that exist between the same Feature-Pair depending on the functional relationship 59

3.12 Catalog of the Feature-Pairs and functional relationships between them 60

4.1 The Drawing Window along with the Main Toolbox 66

4.2 The Feature Toolbox to create Features 67

4.3 The Simulation Toolbox to simulate the mechanical system 67

4.4 The Editing Toolbox to edit parameters of Features and Feature-Pairs 67

4.5 The Assembly Toolbox for Automatic Assembly 67

4.6 Pin handles used to show the select-edit algorithm 71

4.7 Different components before Assembly 74

CHAPTER I

INTRODUCTION

Research Problem

Computer-Aided Design and Computer-Aided Manufacturing (CAD/CAM) have affected virtually every aspect of engineering in modern industry. CAD/CAM helped in obtaining better quality designs of manufactured products. CAD systems are expected to free engineers from tedious, time consuming chores of work that have little to do with technical ingenuity. The present CAD systems speed up designing and manufacturing processes, and strip away much of the tedious paper-work and drudgery that hampers engineering productivity and creativity.

CAD/CAM systems implement computer graphics techniques to improve communication between humans and machine. It is convenient and appropriate to input graphical information to computer and alter output presented by the machine. A dialogue is established through the graphics medium and is termed as interactive computer graphics. In currently available CAD systems, a designer can define and create a part shape, analyze the part, check its technical action, and automatically produce engineering drawings. Further, a production engineer can draw upon geometric description provided by CAD as a starting point to determine process plans, create Numerical Control (NC) tapes, etc. But these CAD systems can only capture an instance of design and have no general basis for managing constraints if changes are made in design.

Feature-based designing techniques overcame the limitations of currently available CAD techniques. Feature-based design implements the concept of features. Features carry information related to both form (dimension, position of feature, etc.,) and function (sliding, rotation, etc.,) of the part. According to Shah and Rogers (1988) the form of a feature is affected by changes in topology of part, while the function of feature is not affected. Feature-based design uses terminology that is meaningful to designers without referring to geometric details of features. Further, application-specific information can be incorporated in the part model from the beginning of design process. Therefore, ideas of feature-based design result in improved designing environment, and hence a better CAD/CAM system.

Feature-based design captures design intent such as assembly topology, product function, manufacturing, etc., while creating part and product geometry . Feature-based design systems make designer think beyond just creating the geometry. Luby et al. (1986) defined features as information sets that refer to aspects of form, function and other attributes of a part. These features can be used in reasoning about design, performance and manufacture of parts. Features are meaningful elements to designers. Features can speed up design process as well as provide a means of standardization, thus reducing cost and time-to-market.

The feature modeller has an integrated data representation that is shared by all designing and manufacturing activities. Features such as block, cylindrical solid, etc., are called as generic features which can be combined to represent various parts of geometry. Application-oriented features such as counter-sunk hole, dovetail, etc., are used in representing a specific application or operation. The feature modeller has the capabilities of defining and managing not only generic features, but also application-oriented features. The feature modeller possesses mechanisms for mapping generic features into application specific features. It has the ability of carrying out consistency verification of geometry and attributes, and also has a versatile user-interface management system. All these mechanisms together confirm the underlying product design and manufacturing methodology.

According to Gardan and Minich (1993), designing is progressive definition of a product, and Feature-Based Modeling (FBM) environment supports it. The purpose of FBM is to construct a feature model along with geometric model. In FBM, any time a feature is added, the earlier feature's characteristic topology is not changed.

Cai (1993) used features to capture the designer's intent for topology and geometry of a single component and has introduced the concept of feature-pairs by establishing links between features. We will extend the concept of feature-pairs by representing functional relationships between feature-pairs which deal with assemblies rather than individual components. The feature-pair based design system captures designer's intent about assembly, early in design phase. The feature-pairs represent the mechanical system at an assembly level whereas features represent at component level. Feature-based design is a special case of feature-pair based design. We will develop a mechanism of representing the relationships between features. Catalog of feature-pairs is the mechanism, which holds functional relationships that exist between feature-pairs. The feature-pairs are represented through a graph structure in the system. The graph structure enables in representing the functional relationships between parts of mating components, relationships between different parts within a component, etc. Since the graph structure can represent these relationships, many applications can be developed using the feature-pair based design technique. The applications are automatic assembly, size and position tolerances of the components in an assemblies, dynamic simulation of mechanical systems, etc. The schematic of the system being developed is shown in figure 1.1.

A feature-pair is a pair of features where one is related to another feature and it is defined with respect to the other feature. Each feature has at least one mating feature and the relationships are represented in feature-pair data structure. When feature-pairs are created, the relationships are automatically determined by geometric reasoner. A compound feature is defined as a group of sibling primitive features. This compound feature is treated as a single entity. The members of a compound feature possesses internal relationships.

Objective

Cai (1993) in his thesis has developed a graphical user interface for the designer to interact with the system. He has also developed a library of features that can be used to represent mechanical systems. He tried to establish links between features through feature-pair data structure. He used the links between feature-pairs to do dynamic simulation of mechanical systems. The objective of this thesis is to start with his work and extend to a proof of concept prototype. The purpose of this thesis is to study the interaction of features in an assembly and the functional relationships that exist between features in components and between components in assemblies. The functional relationships between components is represented using feature-pairs. The software developed is written in C language running on Silicon Graphics. The Silicon Graphics has

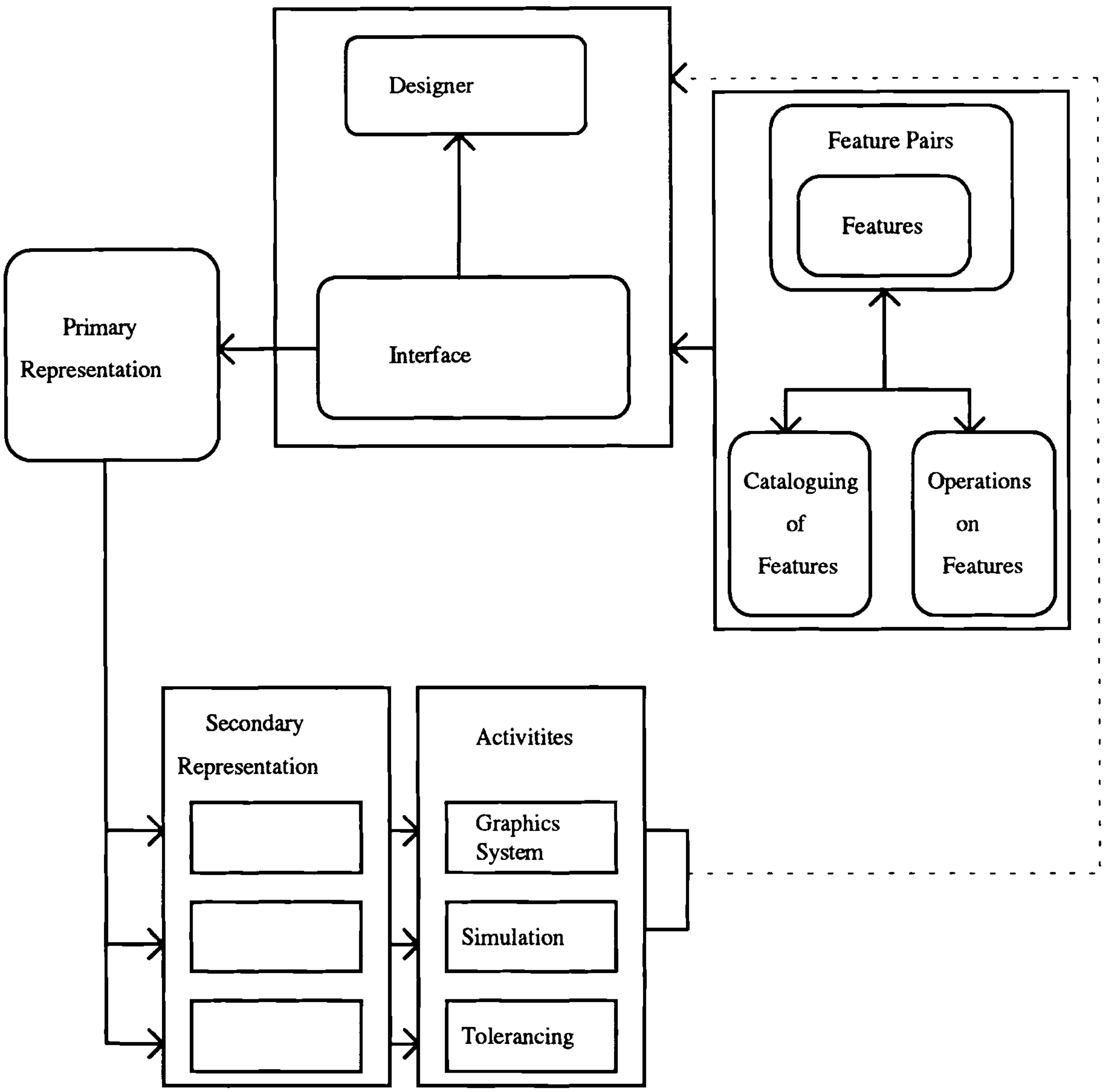

Figure 1.1 Schematic Diagram for the Feature-pair based Design System

the Motif and GL (a graphics library) mixed model, i.e., GLX, and is supporting operating system 2.1. Motif provides a good user interface, while GL provides the required quality and high speed graphics.

The modeling system is developed in 2D space which simplifies the developing work and can handle planar mechanical systems. But since the modeling system is in 2D space, there is some ambiguity in representing 3D objects in 2D space.

From the above discussion, we know that we developed a feature-pair modeling system. The system has a good and friendly user interface to create, modify and delete features and feature-pairs interactively. We developed a mechanism to represent the functional relationships between feature-pairs in assemblies. We tried to catalog the feature-pairs to demonstrate functional relationships that exist between feature-pairs in an assembly or in different assemblies. We will also provide the user with information regarding the interaction of different features of assemblies with respect to the clearances between their components depending on their functional requirements. We developed the mechanism of checking the position and size tolerances in the system. In the system, the designer can opt for rectifying the misalignments in position and size. The data structure supports the mechanisms of catalog, automatic tolerancing, automatic assembly etc. We developed an automatic assembly technique, where we can graphically see the interaction between components during assembly.

CHAPTER II

LITERATURE REVIEW

Overview

Much research has been done in the development of feature-based design systems. In this chapter, we tried to present the background of the design processes and different designing models that are developed. Feature-based design has influenced the designing processes to a major extent. In the section of Overview of features, we tried to define feature and presented the various definitions given by researchers. Feature representation, feature geometry design database and different approaches implemented in feature-based designing systems are explained in detail in the later part of the section. Many feature-based modeling systems are developed and these systems implemented atleast one of the three techniques (feature recognition, design-by-features and interactive feature definition). The techniques that are implemented in the feature-based modeling systems are explained in detail. The functional requirements of a geometrical modeling system for engineering design and applications are proposed by Shah and Rogers (1988) and they are presented. In the later part of the chapter we explained the applications that can be developed using the feature-based design techniques. Lee and Gossard (1985) proposed a virtual-link graph structure to represent an assembly. We tried to study the different applications of feature-based design techniques such as automatic assembly and representation of dimensions and tolerances in an assembly. Fleming (1988) developed a graph structure to represent the dimensions and tolerances of different parts in a component and is explained at the end of the chapter with an example.

Background of the Design Processes

Schulte, et al.(1993) proposed that there are seven working stages in the engineering design process. Engineering design is the sum of all activities which help to work out the information necessary for the manufacturing and the use of a technical product starting from the given requirements or functions the product is supposed to fulfill. The seven working stages proposed are:

1. clarify and define requirements of final design
2. determine functions and their structures
3. search for solution principles and their combinations
4. divide into realizable modules
5. develop layouts of key modules
6. complete overall layout and
7. prepare production and operating instructions.

According to Salomons et al.(1993) prescriptive design models, descriptive design models, and computer-based design models are different models of mechanical engineering designing process as shown in figure 2.1. In prescriptive model approach different design phases and actions to be carried out in each designing phase are

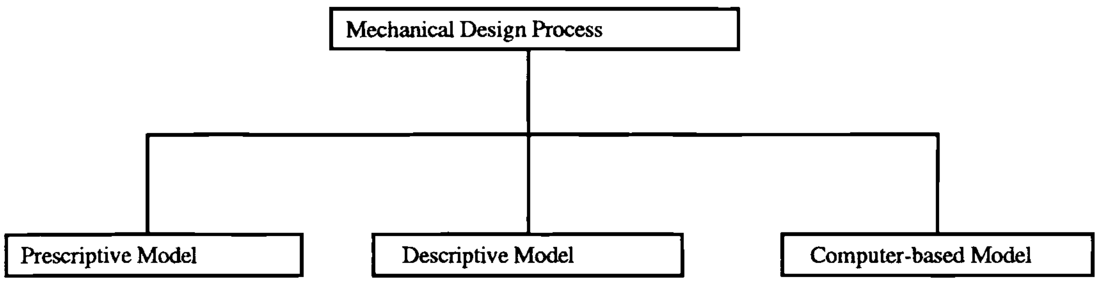

Figure 2.1 Different Models of Design Process

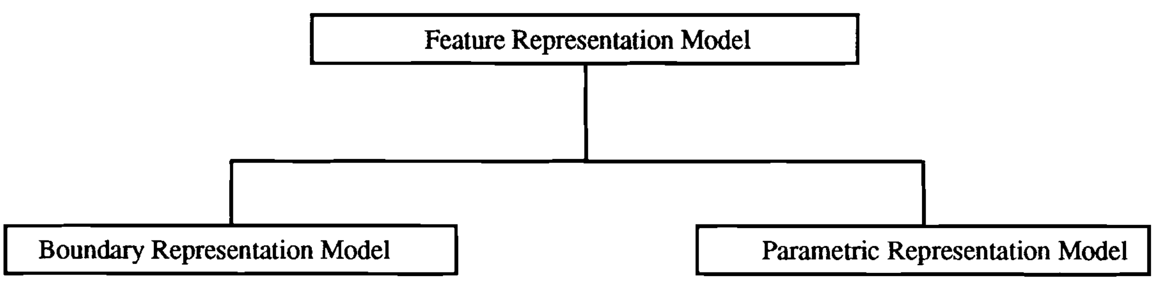

Figure 2.2 Feature Representation in Feature-based Design

distinguished. The prescriptive model approach is a top-down design process. Design phases often referred to, are,

1. conceptual design phase,
2. structural design phase,
3. parametric design phase.

Practicing designers do not follow this model, as they often skip phases or actions in phases.

In an empirical or descriptive model approach of designing process, computer captures the sketches, recognizes design features and builds up geometric models out of the available geometric information. The fundamental geometric attributes or features of design are interpreted by designers from the drawings.

In Computer-based model of design process, interactive computer graphics techniques are used to construct the geometric model of the object. Computer-based models of the design process implements three different design techniques. They are:

1. parametric design,
2. configuration design or structure design,
3. conceptual or preliminary design.

These techniques are explained in later part of the chapter. Feature-based design is a computer-based model. Feature-based design till recently has primarily been involved with parametric design but it is now evolving towards configuration and conceptual designs.

Pratt (1988) proposed that features can be used as vehicles to incorporate functional relationships into CAD systems. Feature has a syntax and is mappable to generic shape. Thus, in representing generic shape of one component, the feature is specific shape element. A feature also consists of specific semantic elements to express feature's engineering meaning. There are relations between syntax and semantic elements of features.

Overview of Features

Definition of Features

Depending on specific characteristics and applications, features have been defined by researchers in different ways.

In a manufacturing based designing system proposed by Cunningham and Dixon (1988), a feature is defined as a geometric form or entity that is used in reasoning about manufacturing activities like manufacturability evaluation, analysis interfacing, tool and die design, inspectability, serviceability etc. In an automatic assembly system proposed by Cunningham and Dixon (1988), feature is defined as semantic grouping to describe a part and its assembly in a relevant manner with functional, design and manufacturing information.

In designing system, the main purpose is to generate process plans. The system implements group technology. Hummel and Brooks (1986) define features as recurring patterns of information related to part description. Luby et al. (1986) defined features as geometric form or entity whose presence or dimensions are required to perform at least one CIM function and whose availability as primitive, permits design process to occur.

Feature is defined by Vaghul et al. (1985) as carrier of product information that aids in design, or communication between design and manufacturing, or between other engineering processes.

According to Salomons et al. (1993), abstract features can be used in designing process, since details of many features will not be known till the end of the designing process. Entities that cannot be physically realized and evaluated until all variables are specified or derived are treated as abstract features.

Feature Representation

Features are significant only when the domain of tools at hand and tasks to be accomplished are clearly specified. Shah (1991) proposed techniques to represent features. The purpose of feature representation is to provide storage, search procedure and a medium for feature-based design as well as reasoning systems. The classification scheme of features must capture topological and geometric variances and invariances among features in a hierarchical manner such as dimensional relationship between parts of a component, etc.

According to Shah (1991), boundary representation model and parametric data model are two ways in which features are represented in a design system as shown in figure 2.2. In boundary representation model, names are assigned when features are created. The boundary representation scheme is capable of handling variations in profile along the length of rectangular block feature, so that end entity (end face) is different from near entity (front face). In a parametric data model, the feature is represented as a set of dimensional and technological parameters, and as a list of edges and relationships. The parameters are used for manipulation of size and position of features in design, manufacturing and process planning. Even in parametric model, features are associated with boundary representation model by using the same names as its constituent geometric entities. Since the boundary representation contains the detailed geometric information, and the parametric data model contains the basic geometric and technological information about features, integration of these two models provide sufficient information for subsequent applications.

The data structure of feature is designed to store geometric and non geometric information in a relational manner. In feature graph, adjacent relationships among nodes and relationships between nodes and linkages are explicitly represented. This information is useful in identifying features that fall into certain pattern and also in finding symmetries for features. For representing mechanical components, face is the basic geometric entity used. Faces of parts can capture great portion of the associated information of parts. A relational data structure captures geometric and non geometric information in representing a feature. The contents of geometric entities are

1. vertex (coordinates of vertex and set of edges intersecting at the vertex),
2. edge (type of edge, parameters to represent edge, start vertex, end vertex, and solid angle of two faces intersecting the edge),
3. loop (type of loop (parent, child) and set of ordered edges and vertices),
4. face (type of face, parameters to represent face, and boundary of face).

The Feature Geometry design database

The need for a design database representation based on empirical data has been discussed by Tikerpuu and Ullman (1987). The database representation provides a structure for describing design objects and constraints in terms of functions as well as forms of features. The changes that are made in a design object as it is refined in a designing process can be recorded. The database representation of features provides a common vocabulary in engineering terminology for designer and relegates the difficulties in defining and understanding design.

The database representation aids design objects in capturing hierarchical relationships between assemblies and their parts. The database representation describes design objects as decomposable units to allow object modifications at all levels of hierarchy. The database provides reference context sensitive information. The designer needs to accomplish the domain specific task. While relating design objects to constraints, the design database facilitates in checking constraint violation and satisfaction.

Different approaches implemented by Feature Based design systems

According to Rimscha (1990) feature-based design systems implement two different approaches in representing geometric model. A **priori approach** starts with abstract notions, and is gradually enriched by geometric and other detailed information. Geometry and topology, assemblies and functions can be modeled using abstract features, mating features, functional features and assembly features. Design-by-least commitment proposed by Mantyla (1989) is a priori approach, in which, exact shape of part is functionally not important. The designer does not make an arbitrary choice but leaves the shape of the part as unspecified.

Conceptual design is a priori approach. Most of the components an object possesses are not known at the stage of modeling. High-level general or specialized functions of the product and principle solutions for fulfilling these functions are determined. Hence features are not related to high level functions.

In configuration design or structure design, which is also a priori approach, the physical concept is transformed into a configuration with defined set of attributes. No particular values are assigned. A general description of a design-object's form and function is obtained based on empirical data from descriptive view towards design process. In configuration design, geometric form of features is not represented, but function is represented. Currently, the central issue is to represent assemblies, i.e., geometry and spatial relations among components. Graphs are used for representing assemblies, where components are represented as nodes in graph, and arcs represent the mating relations between components. The concept and use of features is extended to provide a functionality driven modeling capacity that allows representation of both tolerance and assembly information. Assemblies can be modeled by relating the nodes which serve as geometric points whose position and relative dependencies define components and features. GEKO (GE staltung von KOnstruklionselementen), DICAD (Dialog-oriented Intelligent CAD system), IIICAD (Intelligent, Integrated, Interactive CAD system) are priori approach systems.

The **posteriori** approach is a parametric design approach of a feature-based design. Shah (1991) proposed the posteriori approach and implemented in the system. The nominal shape of a component, its material properties, and variational geometry are defined using form features, material features and precision features. The features define the nominal shape of the component, its material properties and variational geometry. Features have been classified into families, and their properties are identified. Mechanisms

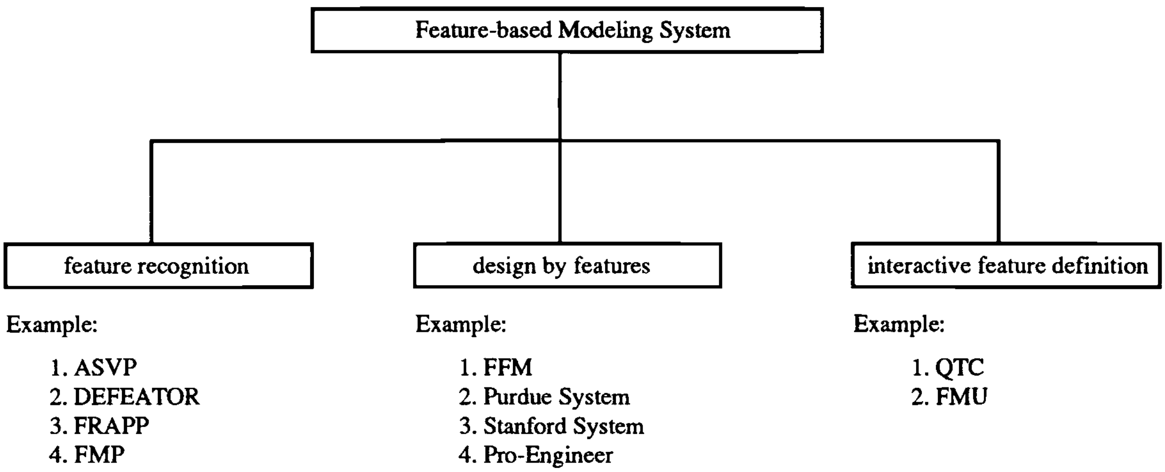

Figure 2.3 Different Feature-based Modeling Techniques

are designed to support each family, instead of having a special method to support each feature.

Wang and Ozsoy (1991) define form feature as a specific configuration on surfaces, edges or corners of a part such as holes, slots etc. The disadvantage with this approach is that although the design intent is captured better than in conventional design modelers, design intent is hard to recover because engineering significance is often implicit to a feature.

In posteriori approach, the designer usually starts with more or less complete geometrical model and defines form features on it. The alternate approach is that the designer starts from scratch by combining form features from a standard library to design a model. Using pre-defined form features in a designing process can reduce the number of input commands. The parametric representation of features is a powerful tool to change the dimensions of features.

Material features include the material composition, treatment and conditions of the form features. These are necessary to obtain the technological characteristics like performance parameters etc.

Wang and Ozsoy (1991) called precision features (called by Shah (1991)) as primitive features. Primitive features are basic geometric entities of a part such as surfaces, edges and vertices, or auxiliary geometric attributes of a part such as center lines and center planes. Form features are built on top of primitive features. These precision features are used to define the dimensions and tolerances in specifying the mating conditions in an assembly description. ASU features test-bed is an example of posteriori approach.

Different feature based modeling systems implemented

There are three types of feature-based modeling techniques as shown in figure 2.3:

1. feature recognition,
2. design-by-features,
3. interactive feature definition.

Feature Recognition:

In feature recognition design technique, portions of geometric model are compared with predefined generic features in the feature library to identify instances that match with predefined features. According to Shah and Rogers (1988), specific tasks in feature recognition include:

1. searching database to match topologic/geometric patterns,
2. extracting recognized features from database, i.e., removing portion of the model associated with recognized feature,
3. determining feature parameters,
4. completing feature geometric model,
5. combining simple features to obtain higher-level features.

Current Problems with feature recognition:

Automatic feature extraction consists of automating the task of determining manufacturing features as part of existing CAD databases such as IGES files, Breps etc. Some of the problems faced in automatic extraction are:

1. Some attributes of the machined part cannot be obtained without referring to a particular feature. When an object is designed without making reference to these features explicitly, it is unclear how to associate machining specifications with proper features. For example, if the datum reference is not specified explicitly, then fixing of jigs and fixtures, positioning of tool head, etc., is difficult.

2. It is difficult to extract features that explicitly depend on the manufacturing process of the component. For example, consider manufacturing of a steel cube. If the cube is to be welded, then the features may be represented as edges of each of six sides to be welded together. If the cube is to be machined from stock material, then the features may be represented as a set of planar surfaces. So, in the designing process, if the cube is represented as edges, and during manufacturing it is treated as stock material, then sufficient information required for machining is not present.

Functioning of Feature recognition process:

According to Shah and Ravi (1992), the designer's features are concerned with functionality, whereas features for which automated recognition processes have been developed are generally concerned with determination of appropriate manufacturing operations in process planning. During the design phase, feature validation and revalidation is required after every modeling operation to check whether a new feature has been correctly installed in the product model, or whether it has destroyed characteristics of some previously created features. Validity checking of this kind requires matching of entities composing a feature, together with their interrelationships against certain rules defining appropriate feature class. In a feature recognition process, the system identifies a set of elements which match defined rules.

According to Henderson and Chang (1988), the feature recognition process has three stages.

1. In the first stage, all simple isolated features are recognized and categorized. Then the corresponding entity loops are removed from database.
2. In the second stage, remaining composite entities are examined to see whether they correspond to partial patterns. Logic is implemented to detect such situations and reconstruct defective features.
3. Third and last stage is gathering all remaining composite entities under the title of generalized protrusion or depression features. This is necessary because no library of machining features can be expected to include all the possibilities which could be created by a designer.

Feature-based design systems implementing Feature recognition technique:

A novel feature recognizer proposed by Kim (1992) provides incremental feature recognition. Feature model representation of a part can be updated on the basis of previously recognized solid-model representation of the part. This is modified by regular solid-modeling operations. Feature models can be manipulated with feature editor by creating new features directly, or by changing attributes of existing features. After such manipulation, geometric model of part is immediately updated. Hence, whenever one of the two representations of a part is modified, other representation is updated correspondingly.

Convex Decomposition of feature recognition uses convex hulls. Convergent Alternating Sum of Volumes with Partitioning (ASVP) decomposition devised by Kim (1992) is a novel approach of recognizing form features which are intrinsic to the shape of model. ASVP decomposition is a hierarchical decomposition of boundary faces of a given model that is based on extremity, where the components abstract boundary face information. The adjacency and dependency relations between faces can be obtained by decomposition procedure. Thus intrinsic interrelations between faces can be systematically found by dealing with components according to the hierarchical structure of decomposition. Combination operations among components of ASVP decomposition are applied on the basis of hierarchical structure and face-dependency information of decomposition. The ASVP decomposition is converted into Form-Feature Decomposition (FFD), where the components correspond to compact and meaningful high-level constituents of shape.

DEFEATOR (DEsign by FEATures editOR) system developed by Unger and Ray (1988), starts with a clean slate and allows user to build part model from scratch. The user is allowed to design new parts from a set of manufacturing features. The user starts with solid prismatic block feature, or a profile feature which is a solid composed of number of connected straight line segments and circular arc segments. The system allows the modifications of features of part model by manual feature extraction, or creation of new part model strictly by the use of features. DEFEATOR can currently load a part model from a file, display the part model, and highlight entities of interest such as features, faces, loops, edges, vertices and points. To select and highlight features, a package of routines is provided to select faces which form holes and pockets.

Feature Recognition And Process Planning (FRAPP) is a feature-based process planning system where a part must be described to process planner in terms of form features. This system has been developed by Henderson and Chang (1988). The function of feature recognizer is to convert geometrical model into form-features automatically. Recognition of features from the model is performed. Feature graph contains local information such as feature type, face-edge lists, sizes of form-features, global information about form-features such as the feature-connectivity and technical data such as tolerance, surface quality and dimension specifications. The process planner uses a feature-graph as input to produce a process plan. Manufacturing criteria and machining specifications are represented using production rules.

At the beginning point of protocol in Flexible Manufacturing Protocol (FMP), developed by Kumar et al. (1988), the designer interacts with CAD system in finalizing a suitable design. The designer checks the design whether it meets certain requirements using enhanced graphics capabilities from the design database which is created in IGES format. The feature extractor decomposes the designed part into a set of standard morphological features like primitive features, i.e., faces, holes, slots, pockets etc. The user is required to specify material and tolerance information which results in a feature file. At this point, fixturing requirements for given shape are determined, and incorporated into an intermediate file. The next step is to evaluate the part and tolerance data in order to determine if the specifications are compatible with the capabilities of machine cells that are available. Then tolerances, geometrical machinability and fixturing constraints are compared with available resources to verify manufacturability of the part.

Design-by-Features:

Shah and Roger (1988) implemented the design-by-features technique in ASU features testbed system. A product model can be built by using (design) features; this is known as design-by-features or feature-based modeling, also known as synthesis by features. Features are functional elements to designers. The central need for developing successful design-by-features system is establishing correct and complete process activity feature sets and capture designer's functional intent from which secondary representations used for reasoning may be constructed. The main requirement of design-by-features system is, it should constitute a natural set of primitives. Complex parts can be designed conveniently with the help of add, modify, and delete operators. Design-by-features enables primary representation of in-progress designs to be created so that desired secondary representations can be developed easily. Feature extraction or decomposition can be computationally tractable.

Inheritance mechanism is implemented by Shah and Rogers (1988) for creating network. The network aids in determining dependent parameters from attributes of other features or algorithmic procedures. Inheritance mechanism helps when a change takes place. The change is propagated to all affected features. Design features often differ from application features. Design-by-features has the advantage of storing relevant information for applications during design process. Design-by-features allows manufacturing and assembly concerns early in the design process.

Two kinds of information have to be considered:

1. feature libraries, which contain generic features (i.e. parametrized features), whose organization is based on inheritance and they are realized by graphs or object-oriented structures,
2. parts and assemblies models use basic features as instanced features to represent parts and various kinds of operations. An instanced feature is an instance of a generic feature and it represents a product model by itself.

In concurrent design techniques, design and process planning are carried out concurrently. This reduces design lead-time, increases opportunities to consider producibility early in design.

Gossard et al. (1988) proposed the concept that design for assembly uses tolerance analysis and group technology to optimize the design. The advantages of group technology is to provide guidelines for designing individual components. The advantage of tolerance analysis is that better assembly can be achieved.

Parametric design rationalizes design process by defining product families with similarities in designing and in process planning, rather than designing many individual products. The geometry and processing of part family depends on a set of well defined control parameters. New designs are carried out by variant approach.

Feature-based systems implementing design-by-features technique:

Form Feature Modeling (FFM) is a technique proposed by Shah (1988). In FFM, features are defined and added to model during design process. Feature recognition is an automated post-design procedure which searches part description, identifies patterns of interest and classifies them according to certain defined rules. Feature identification is an interactive post-design process, in which the user selects entities such as faces, edges, vertices and tolerances and groups them together after naming them.

In the Purdue system, a workpiece is always a rectangular block. In Stanford system, a workpiece is an extrusion of any shape (linear sweep). In Pro-Engineer, a workpiece can be any model created by a linear or rotational sweep. All these systems use a set of predefined features that are subtracted from base solid. In Purdue system, feature model is a list of instances consisting of two levels of information. The upper level encoded information is common to all features, and lower level has specific information. Common data includes the position and orientation matrix, pointers to geometric representations, pointers to reference features, and reference handles. Handles are characteristic geometric elements of features used for positioning and orienting features and for establishing relationships between two or more features. Line handles were used to represent vectors, which included information such as depth of hole or length of slot. Position tolerances are associated with position vectors to locate the position of a feature.

Interactive Feature Definition:

Features are defined by human assistance or interactively. According to Shah (1991), interactive feature definition technique allows the designer to design parts in a convenient way. Interactive feature definition method requires the process engineer to identify machinable features. The identification and extraction of features provides a way for a qualified manufacturing engineer to identify machinable features. But it still does not handle the problem of alternate feature interpretations.

Wang and Ozsoy (1991) proposed a method in which the geometric model of the component is predefined in an interactive feature definition method. The data structure that is used to represent the geometric model of the component plays an important role in an interactive feature definition technique. The component is modeled by the designer using a contemporary geometric-modeling package. A graph corresponding to the hierarchy of topological entities along with the database is created. The database created is read by a program and renders an image of the part on CRT. The image allows the designer to interactively select topological entities (edges, faces) needed to define features. This information can be augmented with attributes such as tolerances and clearances, or high-level nominal parameters, such as hole diameter . This approach is used for inputting data into programs for process planning and NC tool-path generation. In CSG, an object is usually modeled as a binary tree whose leaf nodes are half spaces or primitives, and whose interior nodes are regularized boolean operators. A data structure called VGraph is used for interactive definition of features. The graph contains entities called VFaces (user-defined portions of boundary faces), SFeats (surface features, which are groups of VFaces), VEdges (user-defined subsets of an object's edges), and CFeats (curve features, which are groups of VEdges). Attributes, such as tolerances associated with SFeats, CFeats, and datum systems are defined.

Feature-based systems implementing interactive feature definition technique:

In Multi-model approach proposed by Gardan and Minich (1993), designing using features can be implemented using both interactive and non-interactive techniques. Interactive techniques are difficult to implement but easier to use. The multi-model approach is applied in designing parts and assembly models. Association of features is done to describe parts or assemblies. The part is modeled by the design model of the system and manufacturing model is derived from the modeled part by a set of rules. GGI (Generalized Graphical Input) is used to interactively describe complex constraints. A certain number of values like distances, radius etc., depend on existing geometrical entities. The GGI can be managed by two modules:

1. analysis module for syntactic and semantic analysis of user's actions in order to create an internal model of expression,
2. a valuation module, which evaluates the actual state of model at any given

time.

Quick Turnaround Cell (QTC) is a system developed by Turner and Anderson (1988). Development of QTC is an effort to provide an environment where one-of-a-kind or small batch size parts are designed and produced immediately. The QTC unifies design, automatic process planning, NC code generation, and visual inspection areas into a highly automated production system. Parts are designed in an interactive environment using generic shape features that relate loosely to machining operations. One objective of this work is to incorporate into the feature model, the tolerance information that can naturally represent design intent, and also convey this information directly to process planning. Features are represented as data objects with a list of parameters and identified with key-codes. These parameters describe the feature's geometry, location, orientation and other attributes. The comparison of parameters of different features shows that a number of common parameters exist. These features can be extracted and create feature model.

According to Pratt (1988), each feature consists of two levels of information. First level, i.e., "upper level" contains information that is common to all features with same data structure. Second level, also called as "lower level" has the data which is different for each feature resulting in different data structure.

The Feature Modeling Utility (FMU) developed by Irani et al. (1990), provides an interactive designing environment for defining and manipulating features. The feature representation, in conjunction with geometry modeling utility, Topology And Geometry Utility System (TAGUS) provides tool kit required for feature-based applications, i.e., creating and manipulating features to form a geometrical description of the model. FMU is a layer of utilities which precedes of the TAGUS in the model construction process.

The basic technology and software on which FMU is based are non-manifold topology representational scheme (NMT), which can simultaneously and unambiguously support wire frame, surface and solid modeling. The TAGUS software system, referred to as a geometric modeling utility, bridges gap between modeling systems and geometry dependent application programs. The software system, TAGUS encompasses NMT as its basis. TAGUS is a software system comprising a combination of data structures and operators. The main purpose is to meet the needs of geometry-dependent applications by providing capacity to directly access and manipulate the geometry, regardless of its origin. TAGUS provides a logical bridge between modeling environments and applications, which rely on the modeling environment for their geometry definition. There are three basic components of the system. First, the source geometry which can originate from a CAE-based solid modeller and is transformed into TAGUS representation, second is TAGUS itself and third is target applications which are built with TAGUS operators.

Functional Requirements of Features

Shaw and Rogers (1988) proposed the functional requirements of a geometrical modeling system for engineering design and applications. The intended behavior of the design object is the function of object. Designers generally think in terms of functions before they are concerned with geometry. These functions can exist at different levels of abstraction. In preliminary design phases, functions usually are independent of working principle, whereas, in later design phases, functions are explicitly defined. Then these functions become more and more dependent on working principle.

General functions are restricted in number. General functions are actions on matter, energy and are independent of working principle. Specialized functions are forces, moments etc. Working principle dependent functions are inherent to working principle and are performed by components of assembly.

Working principle dependent functions usually materialize in features that have interface with other components. There are no extensive set of low-level functions at component and feature levels.

The features required depend on the product being designed. The feature modeller must provide an environment for creating, manipulating, modifying and deleting product models. It must be possible to define an instance form feature, precision and material form features individually. Also it must define relationships between form features as well as intra-feature relationships.

Assembly of different components

Ko and Lee (1987) proposed that design objects include assemblies, parts of assemblies, and interfaces between assemblies. These objects are described in terms of context sensitive form and function features. The design object's state is structured as hierarchy of assemblies, components, and interfaces between them. The changes in design object's state are described in terms of operations applied on design objects.

The components are specified by their boundary representation. The relationships between components in an assembly are specified by mating conditions. Currently, four types of mating conditions are implemented:

1. against condition: it holds planar faces of pairs of components,

2. fit condition: it allows rotational freedom of movement of involved components and translational movement along centerline. There are some components with fit condition that do not have rotational degrees-of-freedom between each other.
3. tight-fit condition: force is required to rotate the rod in a hole. If force is great to rotate the rod then this type of mating condition is called a tight-fit.
4. contact condition: this prevents any movement and holds two points of faces of involved components.

Lee and Gossard (1985) proposed that an assembly is divided into several sub-assemblies and each sub-assembly is divided into several groupings. Each grouping is composed of several components. Any two components are said to be in different sub-assemblies if the components have relative motion between them. Any two components in a sub-assembly are said to be in different groupings if the components do not mate directly.

Assembly Representation

Lee and Gossard (1985) proposed a data structure which can be used to represent assemblies in the database. The first part of data structure is used to store topological and geometric information of each component in the assembly. The second part is used to store information about connections between parts in the assembly. In an assembly data structure, two sub-assemblies, two components, or one subassembly and one component are connected by virtual links (basically a graph structure). If more than two components are mutually related, several virtual links can be used so that every pair of mating components occupies one virtual link. A virtual link is a complete set of information required to describe relationships as well as the mating features of the mating pair. An assembly located at top node consists one or more pairs of subassemblies, where every pair is connected by a virtual link. A subassembly consists of several pairs of subassemblies and components which are connected into pairs by virtual links. In this way, terminal nodes of assembly graph will be components of assembly and component data for each component is connected to these terminal nodes.

If many identical parts appear in assembly, data for only one component is stored. By using the concept of instance all the other identical components of the assembly are represented. Virtual links point to instances of components rather than components. Assembly data structure uses boundary representation for each component. Winged-edged data structure with extensions is used to handle multiply connected faces. Topology and geometry are completely separated so that a wide range of surfaces can be handled by modifying data structure of the geometry side only. Small changes to an object cause minor changes to data structure. These local changes in data structure are possible without affecting whole structure. This fact enables easy insertions, deletion of entities such as vertices, edges, and faces. The data structure eliminates the need for searching, which makes most operations fast enough for interactive response. Finally, the data is arranged so that each record is of constant length, which enables effective use of memory.

In the virtual link concept, assembly data is stored hierarchically. The transformation matrix for each component and subassembly need not be assigned, since it can be derived from the mating feature information stored in virtual links. The difficult task is the interactive operation. However, mating feature information can be provided

Table 2.1 ANSI Y14.5 tolerances

Type	Tolerance
Traditional	Size Angle Location
Form	Straightness Flatness Circularity Cylindricity
Profile	Profile of Line Profile of Surface
Orientation	Angularity Perpendicularity Parallelism
Runout	Circular runout Total runout
Location	Position Concentricity
Datum reference frame	

interactively with ease, because mating features are simply graphic elements. Therefore, assembly data can be provided interactively in data structure.

Importance of Dimensions and Tolerances in Assembly

Tolerances, surface finish specifications, and other data that specify allowable inaccuracies or variations from nominal geometry of parts are collectively termed as variational data. These have been extensively studied by Requicha (1983), and Requicha and Chan (1986). Variational classes are families of objects similar to a nominal object and are functionally equivalent. An object is considered to be in tolerance, if its boundaries are constrained to lie within the regions of space called tolerance zones. Currently available modelers provide unambiguous representation of nominal geometry, but lack facilities for representing and manipulating variational information. Size, surface form, curve form, position, surface orientation, surface runout and curve runout are the constraints which constrain the object's features to lie within the region of space called as tolerance zones. Of these size, surface form and curve form are intrinsic to individual features while others are extrinsic. Ranyak and Fridshal (1988), proposed a Table of lists the ANSI Y14.5 tolerances and is shown in table 2.1. The traditional tolerances are basically plus-minus tolerances applied to dimensions.

According to Fleming (1988), tolerances are chosen to manufacture a part so that it can be guaranteed that the part will function correctly despite any variations in shape. During the machining process, a feature may be cut or drilled while the part is being supported by some other feature. Different supports may be used for different cutting operations. Therefore, a feature ends up being positioned with a known accuracy

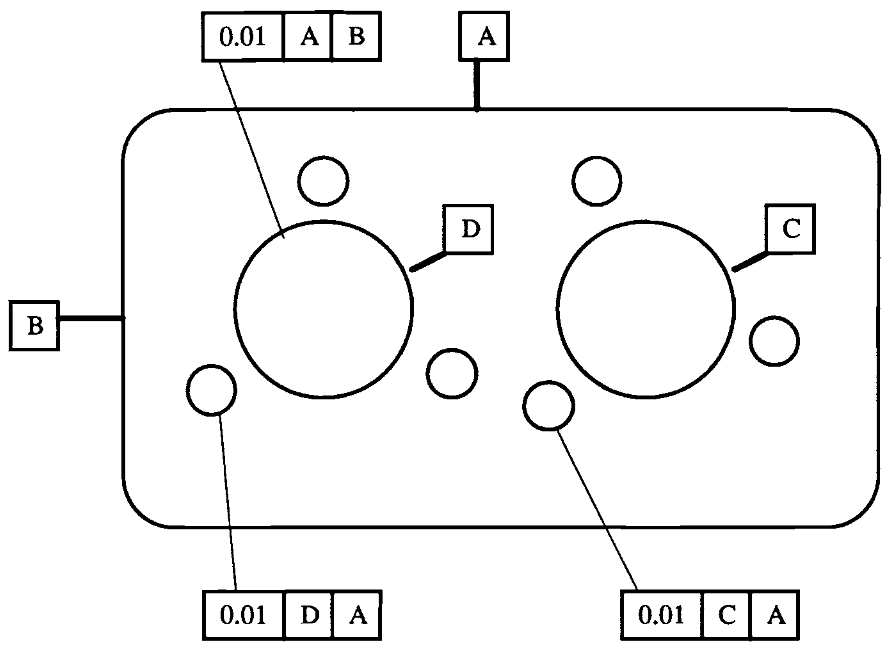

Figure 2.4 Geometric Tolerancing demonstrated by Fleming (1993)

relative to the supporting feature but the accuracy of its position relative to some other feature may not be known directly. These unknown relationships may be important for the satisfactory working of the part. Therefore, it is useful to vary the dimensions within certain bounds. An engineering drawing contains dimensions to indicate sizes of features, distances between features and angles between features. Any dimension may be given a tolerance to indicate an upper and a lower bound for dimension. The problem with this representation is that it is impossible to define unique distance between two imperfect surfaces or when the manufactured part is not properly formed. Techniques called geometric tolerancing enable imperfect form to be taken into account, by using tolerance zones which are regions in which a feature of real part must lie.

Fleming (1988) used the component shown in the figure 2.4 to demonstrate the position tolerances on the component. It is a plate with two groups of four holes. A dial is attached to each group of holes, such that the holes within each group must be positioned accurately relative to one another if they are to meet with holes in the dials. There are three small holes round each large hole and a position tolerance of 0.01 is applied to each. The datums used are C or D, and datum A. The holes have a fixed distance from either C or D and lie at an angle relative to the C or D which is correct relative to datum A. Hence there is a relationship between the position tolerance zone of each hole and each of the relevant datums.

Two main approaches to the definition of tolerance semantics have been discussed by Stewart (1993) and are: perfect-form approach and tolerance-zone approach. In perfect-form approach, the shape and position of object is restricted by means of certain number of constraints on points on the boundary of object. These constraints are typically nonlinear. In tolerance-zone method, boundary of the object should satisfy some sort of slow-variation constraint and hence non smooth form objects are considered and represented using this approach. The perfect form approach is obtained by introducing a very slow variation constraint of the tolerance-zone.

Using the above concepts of features, feature representation, and different design techniques discussed, functional relationships that exist between components of the assemblies are represented. The concepts of feature-pairs is developed to represent the functional relationships. The tolerances and clearances between the components are determined through the data structure developed which leads to automatic assembly of the mechanical system.

CHAPTER III

CATALOGING OF FEATURE-PAIRS DEPENDING ON FUNCTIONAL RELATIONSHIPS

Overview

The sequence of steps to design a mechanical system are:

1. to determine the components in the system,
2. position of each component in the system,
3. the relationships between components of the system.

The designer needs to start with the designing of component. After all components are designed, assembly of these components is done.

In the previous two chapters, the background of feature-based design is dealt in detail. In this chapter, we present the concept of feature-pairs we implemented in our feature-pair based design system along with the importance of functional relationships between components in an assembly. Since feature-based design is a special case of feature-pair based design, more alternatives for creating parts, components and assemblies are given to the user in the design phase. This allows the designer to represent assembly at a higher level.

The mechanism to represent the functional relationships is developed in our feature-pair based design system. A catalog of different functional relationships that exist between feature-pairs is developed. The concept of feature-pairs in an assembly represent the interaction of features in the assembly.

The data structures used in the system to represent features and feature-pairs are dealt with in detail in this chapter. The functional relationships that exist between feature-pairs when they are interacting with each other is explained. Using the concept of feature-pairs we tried to present the different applications (tolerancing, automatic assembly etc) that can be developed in representing the mechanical systems.

Representation of Features in the system

CAD/CAM systems presently available try to support the process of designing assemblies. CAD/CAM systems provide geometric modeling techniques to the designer and share primarily geometric information between CAD and CAM oriented activities. Unfortunately, this approach is inadequate because a large subset of vital design information is non geometric, and many important stages of the design process takes place before geometry is detailed.

Feature is a set of information related to a part's description and the relations that exist between parts. Different kinds of information exist in feature representation which is crucial in designing a part or an assembly. Different types of features exist that correspond to different functions they perform. Form features which we are implementing in our system, correspond to nominal geometry of the part. Precision features represent the tolerances and clearances of the part. Technological features represents the constraints between parts. Assembly features represent the positioning of multiple parts in an assembly and the relationships that exist between the parts.

We use form features in our feature-pair based design system. The form features contains five constituents. They are

1. Solid components: Solid components which are associated with the mass, moment of inertia, etc., and low level geometric and topological information of form features.

2. Measure entities: The measure entities are structures that are used to attach dimensions to the components. The geometric entities and all dimensions that refer to the component are stored in measured entity part of the structure.

3. Size: Size is high level abstraction of specific dimension that controls the intrinsic size of the form feature and refers to measure entities that belong to the same form feature and its value corresponds to the specific parameter stored in the solid component.

4. Location: Locations are used to represent the relative position relationships between mating features.

5. Constraints: Constraints like stiffness, mass, moment of inertia, forces etc., are used to restrict the special behavior of form features.

Hence, the data structure of the object in our system includes the constituents of the form features defined above. The data structure is

```
typedef struct Object{
    int id;
    int type;
    int select_flag;
    int fix_flag;
    double mass, J, vx, vy, ax, ay, theter, w;
```

```
        double *tol;
        double *position;
        int stiffness, color;
        FORCE *forch_head;
        void *field;
        struct Feature_pair *f_pair;
        struct Object *next;
        struct Object *group_next;
        struct Object *group;
}       MY_OBJECT;
```

A pair of axes can be used as datum reference in the system. For every component, after the datum is fixed, the system calculates the relative positioning of each component in the system relative to the datum. The relative distances are stored in the position field of the data structure.

We use implicit feature definition in our system. An implicit feature definition is an unevaluated definition where minimal amount of information, i.e., extremely compact, is stored. Each of the feature classes are defined in terms of rules and sets of parameters. The assignment of specific values to these parameters result in an unevaluated feature which in principle is associated with a particular position and orientation of part. Evaluation of these implicit features lead to explicit type of representation of these features. All features are represented implicitly until operations are called for. These operations demand the evaluation of information of implicit features. Evaluated features are not stored in the system. Then the evaluated features could be deleted from the model after the model is generated with the information. The implicit feature definitions are defined by linear or rotational transformation.

Now, consider the case of rectangular block and a cylindrical solid which are features in our system. The data structure used in representing the features is shown below:

```
typedef struct p3d{
        double *point;
        struct p3d *next;
}P3D;

typedef  struct  block{
        double *center;
        double *dir;
        P3D *polygon;
}MY_BLOCK;

typedef struct pin{
        double *center;
        double r;
        P3D *polygon;
} MY_PIN;
```

Pin is a feature and dimension is the specific information of the pin, i.e., the radius which is fixed. The position of the pin is determined from the center location. This gives the form details of the feature. Similarly in the case of the block, the data is stored in the form of intrinsic feature representation. In the actual construction of the geometry, the explicit details of constructing the geometry are calculated. Using these values, the geometry is constructed. For example, in the case of block, location of center, the length, width and angle of rotation is stored. When the operation for drawing the block is called, then using the stored implicit data, the corner locations of the block are calculated.

As far as the representation of form features is concerned, it is the same for both shaft and pin. Only when it comes to the function of the application feature, the representation of features is differentiated. This is explained in detail in the next section.

Concept of Feature Pairs

In the previous section, we discussed how features are used to represent the components in an assembly. Any given mechanical system consists of many components, and the relationships between components interacting in an assembly should be represented properly using the data structure. The data structure is divided into two parts. The first part is to store the topological and geometric information. The second part is to store information on how all the components in an assembly are related. Feature-pair data structure is a better approach of representing and implementing assembly in the design system.

The features capture the designer's intent in component design. The feature-pair captures the designer's intent about assembly in the design phase. Feature-pair is a set of information to describe the mating conditions that exist between two mating components like pin-hole, pin-slot etc.

In the virtual-link data structure proposed by Ko and Lee (1987), information regarding the assembly cannot be given to the system during the design phase. The data structure used in the system has information in the geometric level which is not useful when representing constraints like tolerances, etc. These constraints should be represented at feature level. We have overcome the difficulties of virtual-link data structure proposed by Ko and Lee (1987) by using another methodology i.e., feature-pair based design in our system. A single feature can only describe the topology and geometry of itself. It is not enough to describe an assembly. Hence we propose the methodology of feature-pair. After analyzing the virtual-link data structure, we found that a virtual link is actually a pointer which connects the nodes in tree structure. It is possible to change the content of virtual link without changing the hierarchical tree, so that the changes in feature information is included. Next, we will develop the concept of feature-pair, which works just as virtual link. We will develop a feature-pair data structure and a graph structure to represent an assembly in a hierarchical tree. The layout that is proposed looks almost the same as that of Lee and Gossard's (1985) hierarchical tree. In feature-pair data structure the information is stored at feature level whereas in virtual-link data structure the information stored is at geometric level like face, edge, etc. We build up a higher level structure on the basis of feature-pairs in the feature-pair based design system. This provides a powerful tool to the users and allows them to represent the assembly at a higher level.

In today's CAD system, the designer needs to start with the component design and then later when the components are designed assembly is generated from these components.

Consider the assembly with a sliding pin in a slot. Initially, the components are created. First with a block as shown in the figure 3.1(i). Later a slot is made in this block as in figure 3.1(ii). Next in another block a pin is attached as shown in figures 3.1(iii) and 3.1(iv). Now, an assembly is generated by joining these two components as shown in figure 3.2. The data structure for this method of assembly of components is shown in figure 3.3. The two individual groups of components are connected by slot and pin. This type of representation is useful to represent the components that form groups in the overall assembly. But this way of representation doesn't give the interaction details of features in the assembly. Hence we now need to move this graph structure to an assembly modeller to represent the assembly.

Now let's discuss how the feature-pair based design represents the assembly. By creating the feature-pair of pin-in-slot, we create a feature-pair data structure to automatically hold the two features and their relationships. This is shown in figure 3.4(i). The most important is that we represent the assembly relationship early in the design phase. Now, the slot is attached to a block and the pin is attached to another block as shown in figure 3.4(ii). The data structure is shown in figure 3.5. The difference between the CAD system and the feature-pair based design system is clear and we see that the feature-pair works just as the virtual link (links two components together). The design procedure is almost the same, except that we represent assembly when the components are automatically created at the design phase.

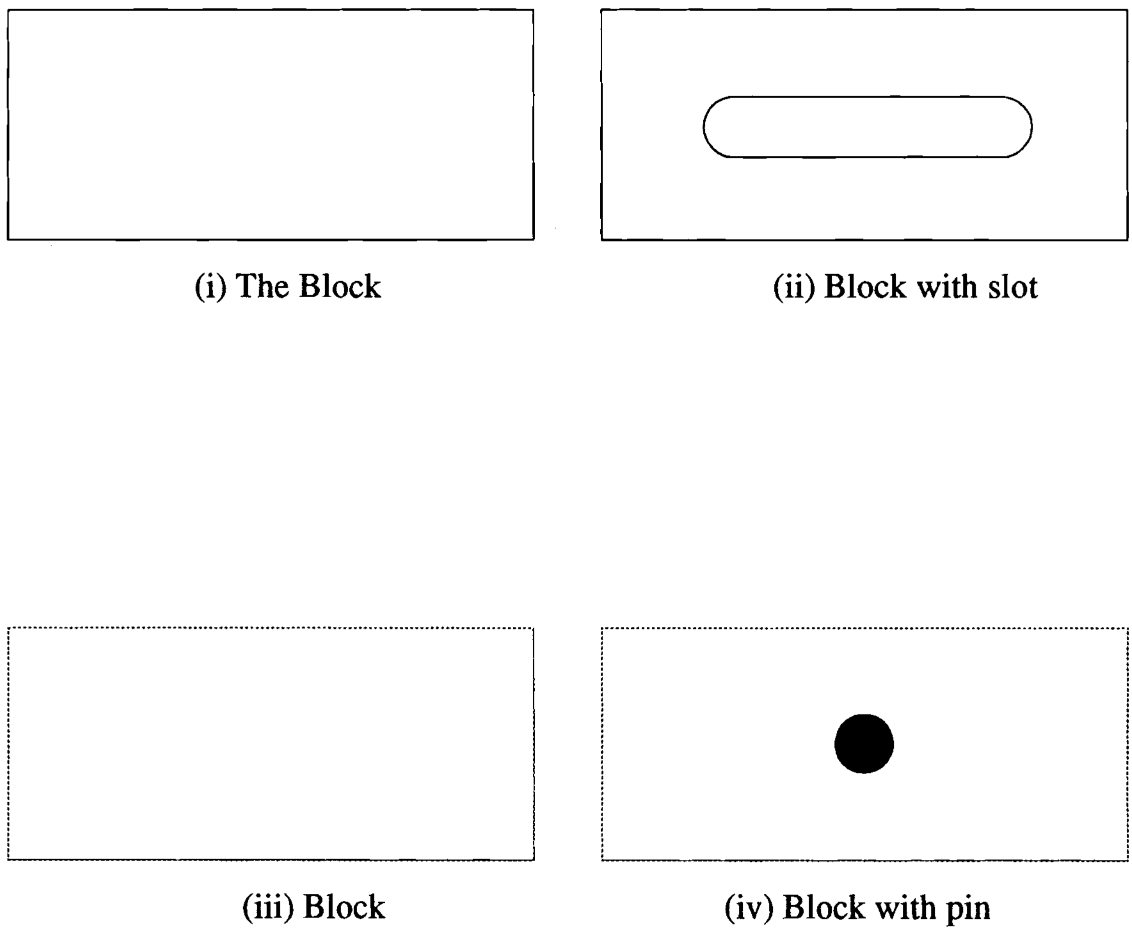

Figure 3.1 Different components before assembly (block-with-pin and block-with-slot)

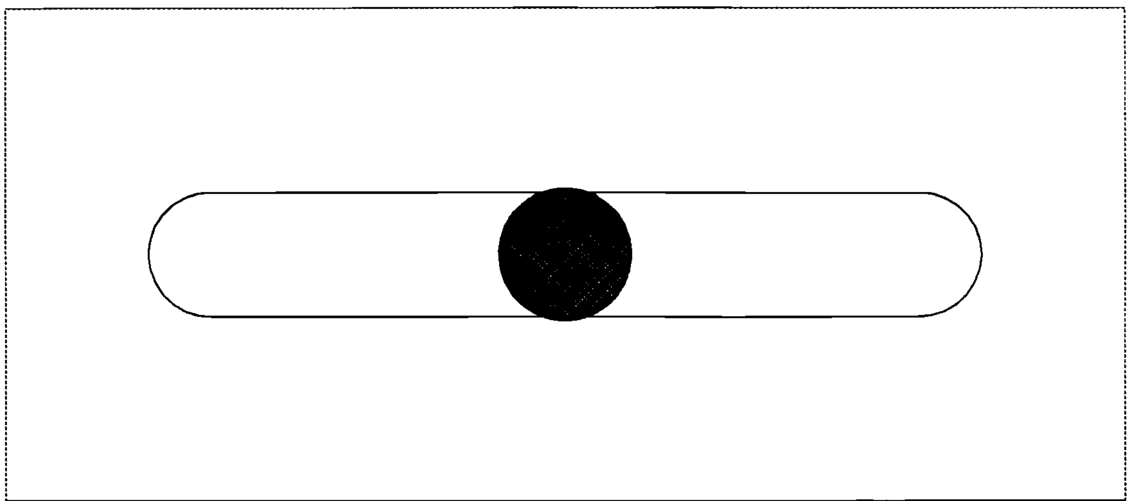

Figure 3.2 Assembly of pin in slot

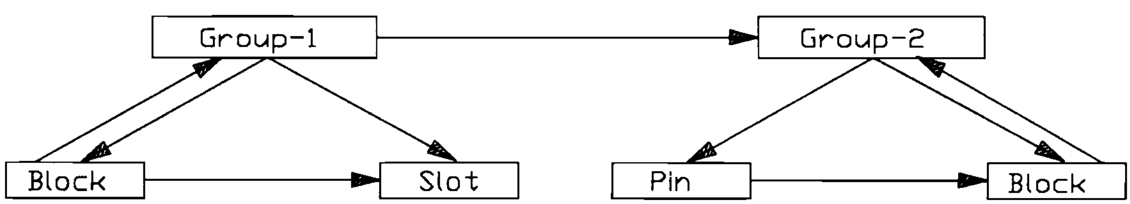

Figure 3.3 The data structure used to represent the assembly of components

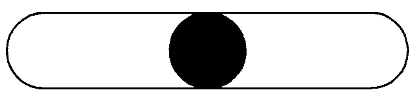

Figure 3.4(i) feature-pair

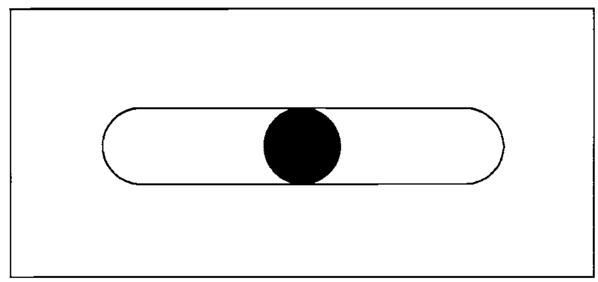

Figure 3.4(ii) Assembly using feature-pair

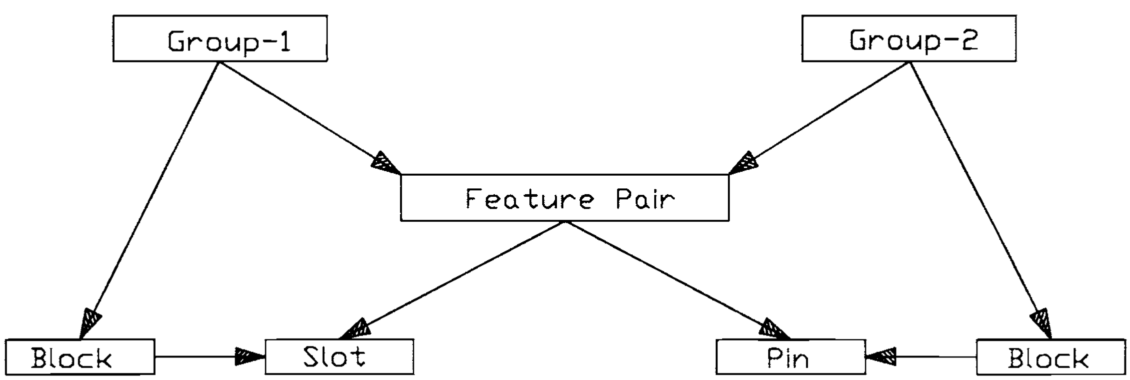

Figure 3.5 The data structure to represent the assembly using feature-pairs

We define the feature-pair data structure as follows:

```
typedef struct feature_pair{
        int id;
        int type;
        int clearance;
        MY_OBJECT *obj1,*obj2;
        struct feature_pair *next;
} F_PAIR;
```

As mentioned before, the object data structure is a hierarchical tree. The feature-pair points to the feature in a component, and not the component itself. If we traverse up in the hierarchy from the feature , the top-level group of it would be the component where the feature attaches.

In figure 3.6, various feature-pairs have been used to design the mechanism. The mechanical system shown in the figure 3.6 is an assembly of components like slider, slots, pins, links etc. The slider in the slot moves in the slot as the mechanism operates. Since these two, i.e., slot and the block (slider) are defined as a feature pair within the assembly, any changes made to the slider will redesign the slot according to the changes that are performed. This type of designing facility aids the designer in making good designs as the system automatically makes the changes in the related parts of the assembly, if a change is made in one part of the assembly.

In the latest approach of feature-pair based design, the initial stage of designing a system is the assembly. Consider the hole and pin, which form a feature pair to represent two components of an assembly. Since two features of feature-pair know their relation between each other, we can use the feature-pair concept to represent the relationships that

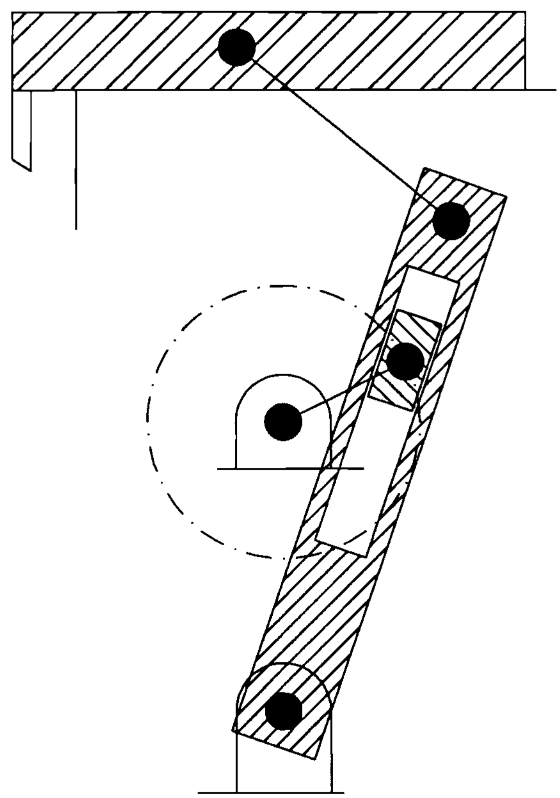

Figure 3.6 Mechanical System involving slider-crank mechanisms and pin joints

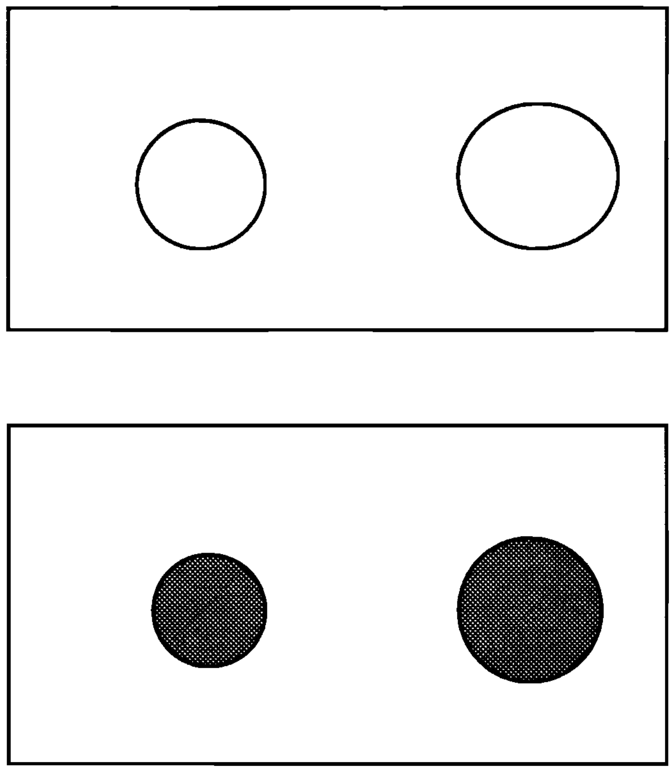

Figure 3.7 Two components of the assembly which consist of mating feature-pair

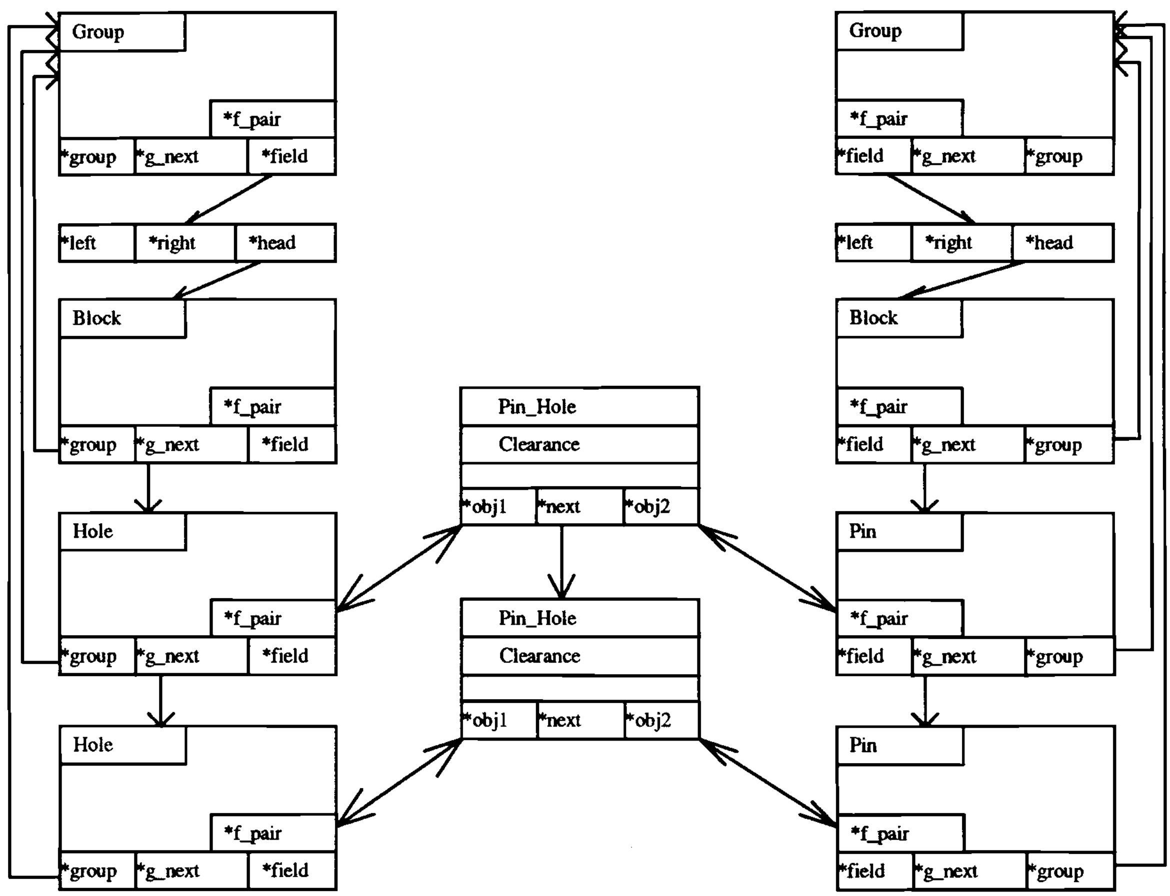

Figure 3.8 The tree structure implemented in the system to represent assemblies

exist between the two components. In figure 3.7, the mechanical system has two different components. The hole on one component is related to the pin in another component. Feature-pairs are used to represent the relationships between the components of the assembly. If the designer analyses one of the components and redesigns the parameters of hole in one of the component, then the corresponding changes are made in the mating component of the assembly. This helps the designer in making quick and reliable designs. In figure 3.8 the tree data structure that is being implemented in the system to represent the assembly is shown. From the structure, we can see that there are links between different parts in individual components and also there are relationships between the features of mating parts through feature-pair links. This type of structure gives an additional flexibility in establishing relationships between functionally related components.

Tolerances and Clearances in an assembly

A major deficiency in currently available geometric modeling systems is they lack facilities for specifying tolerancing information, which is essential for design analysis, process planning, assembly and other applications of modeling. Information regarding tolerances and clearances is crucial for designing and manufacturing of products.

Dimensions are natural descriptors of geometry. Dimensions are logical and appropriate "control points" to alter a component's geometry. Dimensions are one among the important designing or manufacturing considerations. Tolerance specification is a geometric constraint on an object's boundary features. Clearance specifications are the constraints on mating features. An object is in tolerance if its boundaries lie within tolerance zones. Tolerance zones are regions of space constructed by offsetting (expanding or shrinking) the object's nominal boundaries.

Both dimensions and geometric tolerances are defined with respect to basic constituent features of components. Sub feature level information like conceptual entities (axes, median planes, central lines) are explicitly represented. Although the basic primitives are similar to those defined in parametric system, axes of basic features are represented distinctively and their names are internally consistent. The faces and axes of the basic features can be toleranced or can be used as datums. This representation scheme can be viewed as an extension of parametric approach to include edge and axes information for tolerance definition and process planning.

Tolerances, surface finish specifications, and other data that specify allowable inaccuracies or variations in nominal geometry of parts are collectively called as variational data. Tolerances constrain an object's features to lie with regions of space called tolerance zones. Different types of constraints exist like the size, surface form, curve form, position, surface orientation, surface runout and curve runout. The size, surface form and curve form are intrinsic constraints and depend only on individual toleranced features. Position, surface orientation, surface runout and curve runout are extrinsic constraints. The extrinsic constraints are defined relative to other features and requires datum specification. The fit between a shaft and hole must fall into one of these categories, clearance, transition or interference. If the shaft is smaller than the hole, they will assemble easily, and this will be a clearance fit as shown in figure 3.9(i). If the shaft is larger in diameter than the hole, they will be an interference fit, and certain amount of force will be necessary to fit them together as shown in figure 3.9(ii). Between the interference fits and clearance fits lie a range of fits known as transition fits. These are obtained when the upper limit on the shaft is larger than the lower limit of the hole This is called as transition fit and is shown in figure 3.9(iii).

In the system, we tried to implement the size and position constraints on the objects and object's boundaries. Size constraint is the size tolerance of the component. The tolerances on each component are calculated using the following relation:

$$i = (0.052 X \sqrt[3]{D} + 0.001 X D)$$

where D is the geometrical mean of the two diameter limits. The 0.001D is the uncertainty measuring factor which increases with diameter. International Tolerance Grades are designated by the notations IT1, IT2, ... IT16. Each grade consists of tolerance values that increase with the magnitude of dimensions. These are fixed depending upon the precision attainable with the various manufacturing processes. We have fixed the IT8 as the default in our system to calculate the permissible tolerances of the components in the system.

Position tolerances are incorporated in the system to check whether the interacting features within a group are properly aligned with respect to the datum. From the data structure shown in the figure 3.8, we see that there is a loop existing between the interacting features of different components within an assembly. Using this loop, the positional relations between the components are checked. The true positioning is shown in figure 3.10(i), 3.10(ii) and 3.10(iii).

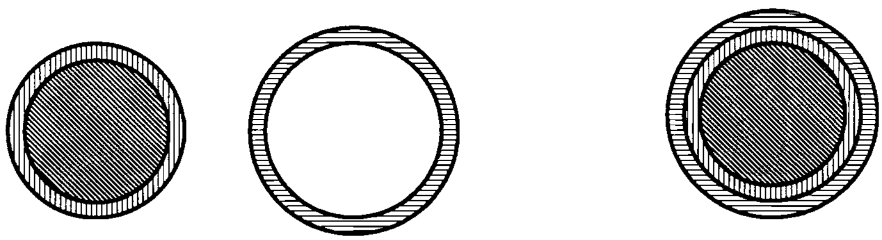

Figure 3.9(i) The clearance fit

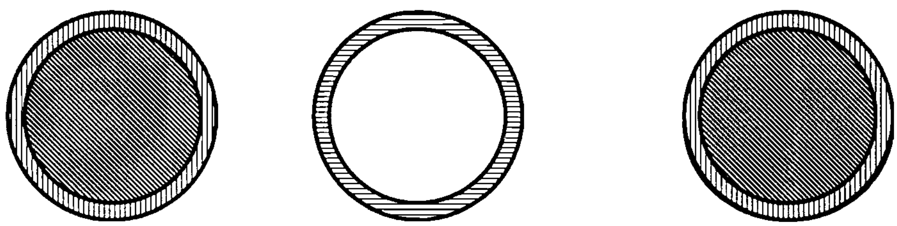

Figure 3.9(ii) The interference fit

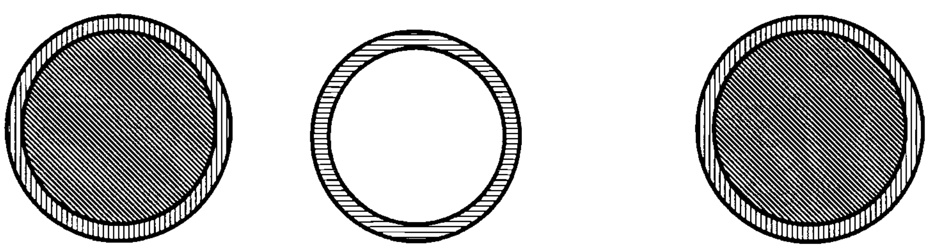

Figure 3.9(iii) The Transition fit

Dimension-Driven Geometry (DDG) refers to computational methods for automatically translating changes of dimensions into corresponding changes in geometry. The component geometry can be rescaled by altering dimensions proposed by Roy and Liu (1988), and Ranyak and Fridshal (1988). Hence, DDG provides a natural, rational and efficient method for modifying geometry and hence provide an explicit means of representing dimensions, tolerances and form features in geometric models. Such explicit representations will provide an important foundation for higher level application programs to automate design of assemblies and to automate

tolerance analysis and synthesis. A feature-pair is selected from catalog with given clearances, eg. pressfit is selected, and the features are placed in different components. The designer can edit the dimension of the feature-pair. One of the features of the feature-pair is selected and its dimensions are changed, then the corresponding changes take place in the mating feature, but still maintain the mating constraint of pressfit.. This is one of the useful techniques in making design changes quickly.

Between same feature-pair, depending upon their functioning, the tolerance limits and dimensions vary. For example, in the component shown in figure 3.11, when there is only translational motion between the slot and pin, then the tolerance limits applied are different compared to the rotational motion of slot about the pin.

Features have many functions associated with them. For example when the shaft is interacting with a hole it has different clearance limits when compared to the same hole with a pin going into it. This functional dependency between the different features like shaft and hole, etc., are represented through the catalogue and the user is given a option to select the type of constraint he requires. If the designer prefers to have a pressfit after the designing instead of the loose-fit which he defined while defining the assembly, he can

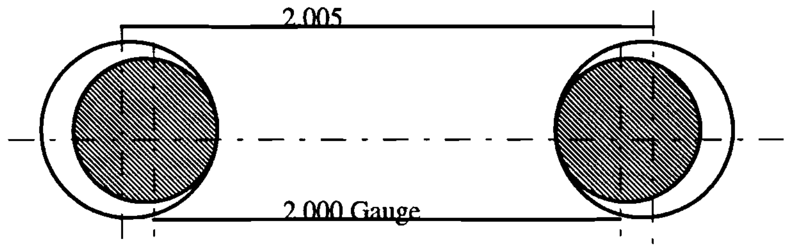

Figure 3.10(i) When the holes are far apart

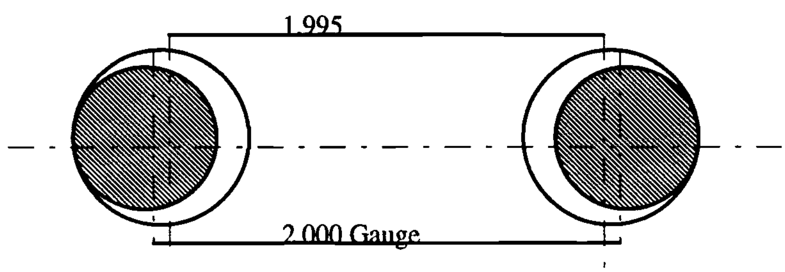

Figure 3.10 (ii) When the holes are closest

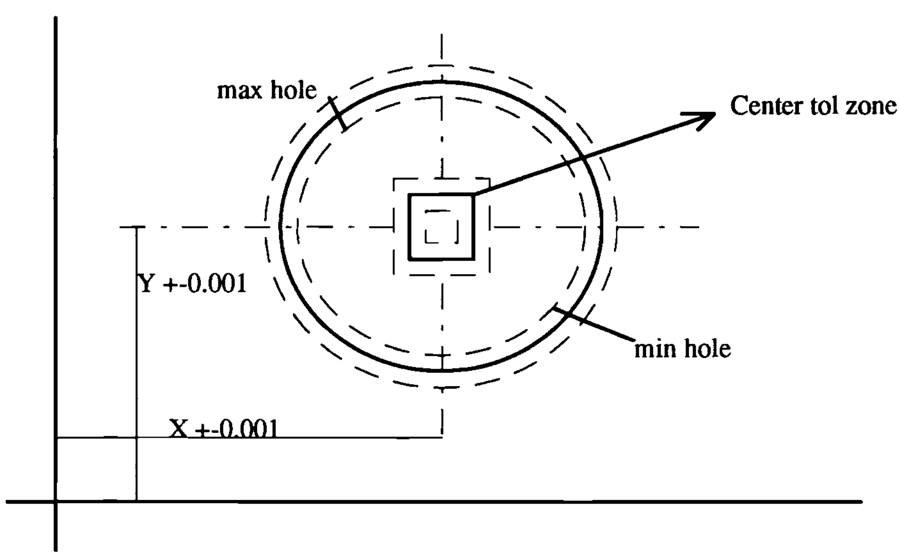

Figure 3.10(iii) True positioning with respect to datum axes

change the constraints and parameters at that instant by changing the clearance relation between the feature-pair. Implementation of this technique in our system, is being explained in detail in the following chapter.

Cataloging of Feature pair Depending on Functional Relations

Different types of functional relationships exist between features depending on their mating conditions. For example, a feature i.e., cylindrical solid acts differently when it is used as a shaft when compared a pin. This relationship is determined depending on their functionality in the assembly. Cataloging is one of the better methods of presenting the functional relationships so that the designer is aware of the existing relationships between the features, early in the designing stage.

Cataloging of feature pairs aid the designer in selecting the feature-pairs for the assemblies. Depending on the requirement of functional relationships between different components in an assembly, the designer can use the catalog to select feature-pairs. Through the catalog, the designer can know the different types of relations that exist between different feature-pairs provided in the generic feature library of the design system. Depending on the requirement, the designer is given a chance to choose the feature-pair to design a mechanical system. The features are put in form as shown in the figure 3.12. The designer in the designing phase requires a hole and a pin relation between two components of an assemblies. Then, the designer can use the catalog to find the different functional relationships that exist for the feature-pair and can choose from there the required relation, i.e., either a shaft and hole with loose-fit , pin joint with a press-fit etc.

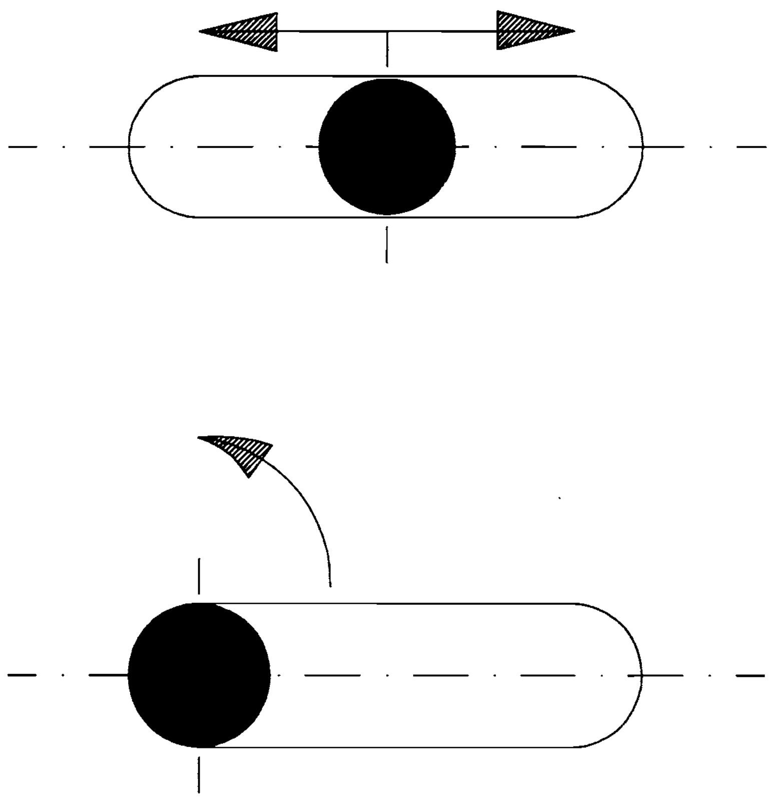

Figure 3.11 Showing the different relationships that exist between the same feature-pair depending on the functional relationship

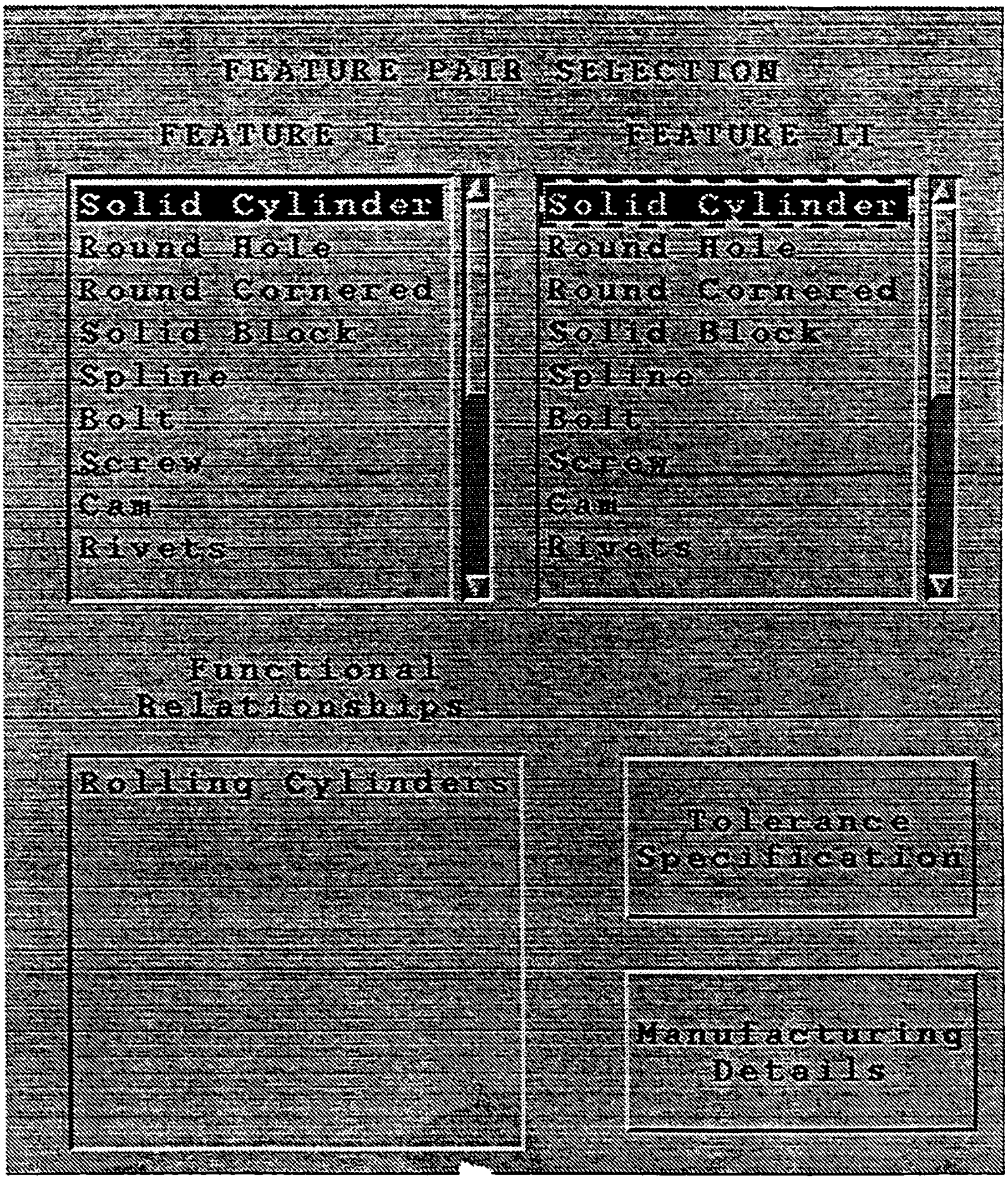

Figure 3.12 Catalog of Feature-Pairs and Functional Relationships between them

Manufacturing Details depending upon the Functional Relationships of Feature-pair

In an automated designing and manufacturing system, it is desirable to provide the designer with natural designing environment, and to provide the process-planning system with as much detailed machining information as possible. Feature-based design approach is one of most popular approaches to the development of such designing system. The envisaged level of sophistication of such systems is that they should provide the designer with a rich vocabulary of features. This helps the designer in specifying the geometry of designed part.

Some attributes of a machined part cannot be made without reference to a particular feature (example, the surface finish, corner radius and machining tolerances of a pocket). When an object is designed without making reference to these features explicitly, it is unclear how to associate machining specifications with the proper features. In the case of machined parts, one problem with the design-by-features is that it requires a significant change in the way a feature is designed. Traditionally, a designer designs a part for functionality, and a process engineer determines the manufacturable features from the part. However, the design-by-features approach places designer under the constraints of not merely having to design for functionality but also having to specify all of the manufacturable features as part of the geometry--a task the designer is not normally qualified to do.

Process planning deals with selecting and defining the processes that have to be performed to transform the material into a given shape. The decisions made in process-planning relate to single parts. Process planning includes:

1. interpretation of product model,
2. selection of machine tools,
3. selection of tool sets,
4. selection of setups,
5. selection of machining operations and their sequence,
6. selection of cutting tools,
7. design of jigs and fixtures,
9. calculation of cutting conditions,
10. determination of tool paths,
11. NC part-program generation and
12. capacity planning.

Manufacturing features are key to generate the process plan. There are two reasons for this:

1. manufacturing features provide for a natural form of communication;
2. process planners think in terms of holes, pockets etc., and manufacturing features simplify process planning since there are only a finite number of ways to manufacture a feature.

Process for manufacturing of components depends on function, the component has to perform. This can be decided by the designer at an early stage using the catalog that is developed. If the designer needs the feature to perform certain functions, then the manufacturing process varies depending upon the function. Consider the case of a feature-pair, cylindrical solid in the round hole. If the cylindrical solid is to perform as a shaft then the process of manufacturing it is different from that of a pin. The tolerance grade for the feature can be fixed, depending on which the clearance limits vary. This is one of the possible application of the feature-pair based design system.

CHAPTER IV

IMPLEMENTING THE FUNCTIONAL RELATIONSHIPS IN THE SYSTEM

Overview of the system implemented

The system is designed for defining two dimensional mechanical systems in a general way. A graphical user interface is provided for creating, editing, and deleting features on drawing area window. The system is designed in direction of the object-oriented methodology using callback functions. Object-oriented method permits the features to be added to the system or deleted from the system, without influencing other features in the system. This methodology is useful and there is a clear hierarchical relation between the features that make a mechanical system.

In most of the traditional designing systems, once the application has started, the interface is in control of the application. The interface allows only certain kind of information to be input into the system by the designer. For example, the application might ask the user to input through a menu and use the reply of the user to go down a level to a new menu. Here the actions that were possible at the previous level are no longer available. Or a text editor may operate in one mode in which keyboard input is interpreted as editor commands and in another mode, it is interpreted as data that is to be stored in an editor buffer. In any case, only keyboard input is expected.

We implemented event driven algorithms in developing the user interface of our system. In our system, multiple graphic applications can run simultaneously. In an event driven system like this, the user can use keyboard, the pointer to select data, click on buttons or scroll bars. The designer can change the keyboard focus from one application window to another by moving the mouse to other application window. The user can suddenly switch from the keyboard to the mouse, or from one application area to another. Furthermore, as the designer moves and resizes windows on the screen, application windows may be obscured or redisplayed. The application is prepared to respond to any one of the different events at any time. If the user is in editing window then the focus is on that window and the icons are highlighted so that the user can select items from that application window.

Event driven programming reduces modes to a minimum. Hence, the designer need not navigate a deep menu structure and can perform any action at any time. The designer is in control of the designing process and not the application. The application simply performs some setup and then goes into a loop from which application functions may be invoked in any order as events arrive.

The User Interface

In our event driven designing system, the graphical user interface opens up with two application windows. One is the main window for displaying objects. The other application window is a tool box, which contains five icons (features, catalog, editing, simulation and assembly) in it. This is shown in figure 4.1. Clicking an icon of the tool box activates a callback function which pops up a window and remains open until the

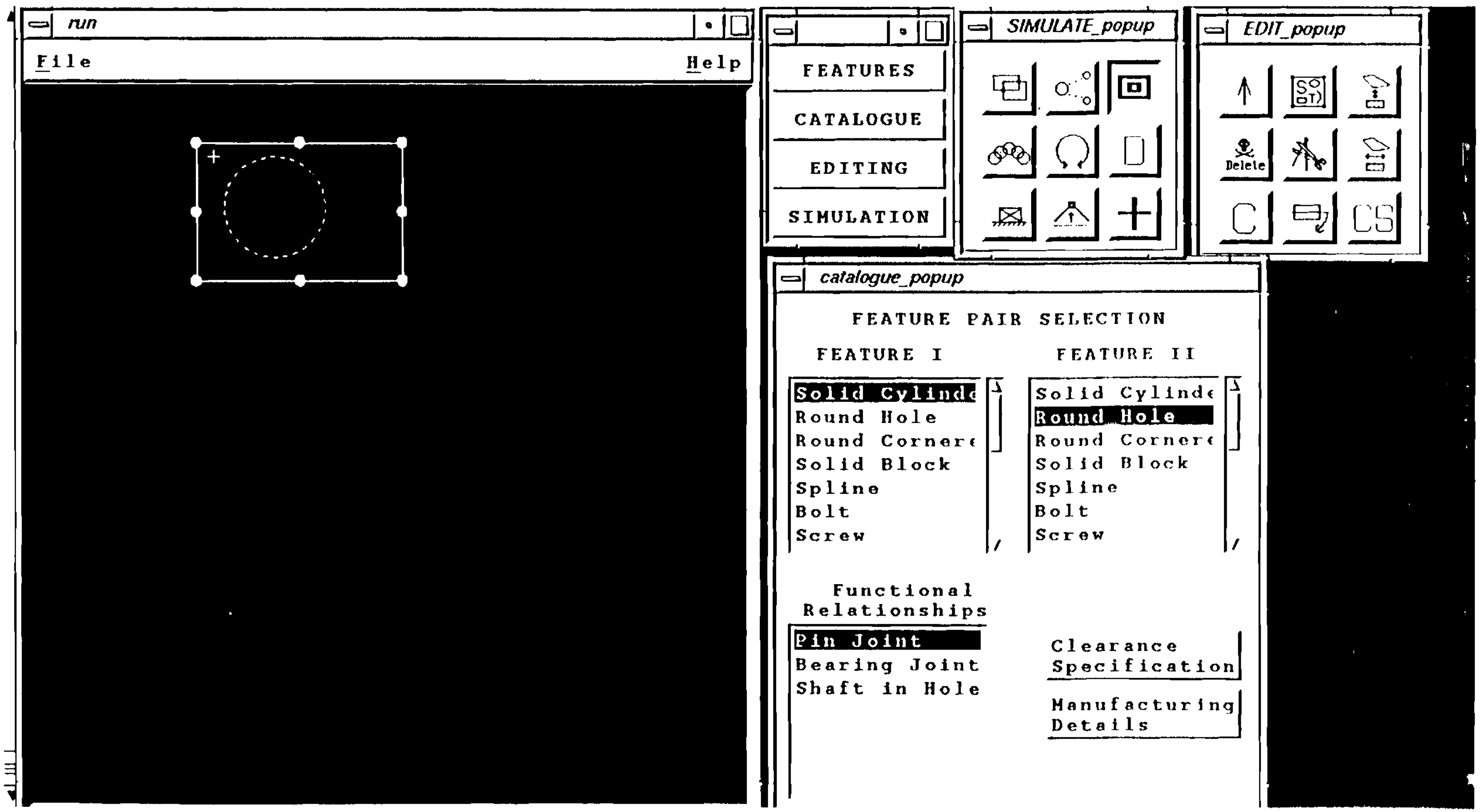

Figure 4.1 The Drawing Window along with the Toolbox

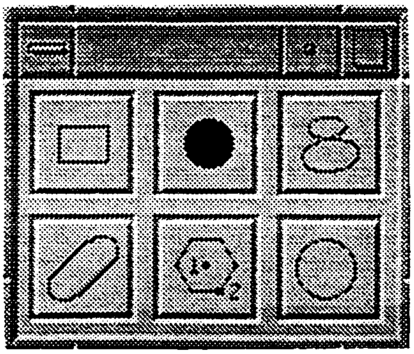

Figure 4.2 The Feature Toolbox to create Features

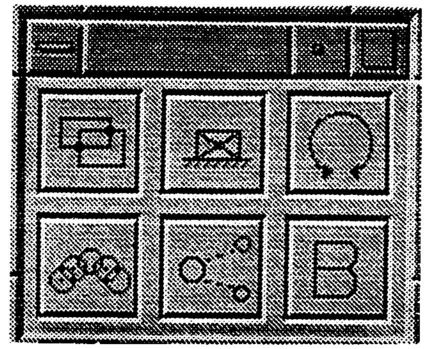

Figure 4.3 The Simulation Toolbox to simulate mechanical systems

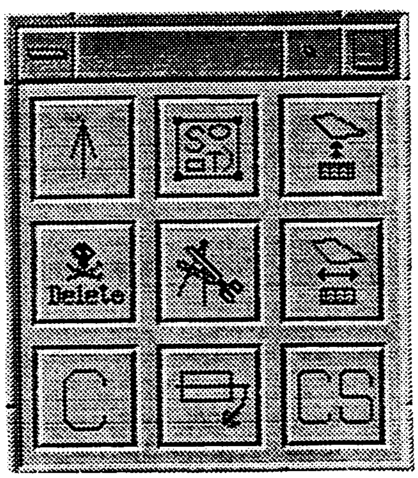

Figure 4.4 The Editing Toolbox to edit parameters of features

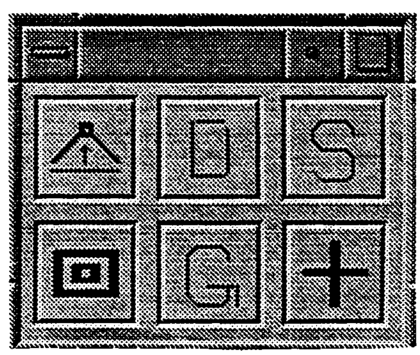

Figure 4.5 The Assembly Toolbox to for automatic assembly

designer closes the window. When the icon for editing is clicked with mouse, then a window with the editing icons pops up. Similarly, the features, assembly and simulation icons. The designer can use the different options (different icons for selecting, deleting etc.,) of the created application window in editing the created objects. When an icon in the editing application window is chosen, a callback function corresponding to the icon is called and the system interprets mouse action according to the current mode. All parameters about features are interpreted by the system through the mouse action. The features tool window has different options of creating individual features like the hole, pin, slot, block etc., as shown in figure 4.2. The simulation application window has the tools for checking the interference between features, simulation of the system etc., and is shown in figure 4.3. The editing application tool window has the options for editing the existing features dimensions, positions, deleting the features, grouping the features, etc., and is shown in figure 4.4. The assembly application window has the tools for checking the positioning of features in a component, viewing the feature-pairs of the system, showing the assembly etc., and is shown in figure 4.5.

We developed a Catalog of feature-pairs as shown in figure 3.12. In this catalog, two columns of feature lists exist. The designer selects one feature from each column to form a feature-pair. After selecting two features of the feature-pair, the functional relationship between the feature-pair appears on the scrollbar window below the feature lists. The designer can select the functional relationship from the scrollbar window. Then, the feature-pair accompanies the pointer of mouse on the drawing window area. The designer can click on the point where he wants to on the drawing area and then drag the mouse to resize the dimensions of the object. For example, the user selects solid cylinder and cylindrical hole as two features of the feature pair, then the relationship appears in the functional relationship window below the two column lists of features. Once the designer clicks on the pin-joint with pressfit, then if the designer clicks on the drawing area window and drags with the mouse on the drawing area to resize the dimensions of the feature-pair. The selection of feature-pair using the catalog can be done in this way.

The functional constraints like the clearances and tolerances are assigned default to the features of feature-pair depending upon the functional requirements and the dimensions chosen for the features. If the designer wishes to change the constraints, then the designer needs to click on the button for tolerance specification of the window. A small window pops up showing different clearances and fits that are available in the system like pressfit, loosefit etc. For example, when the user chooses the round hole and solid cylinder as the two features of feature-pair, then the functional relationship appear on the window. The user wants a pin joint with pressfit at some location as shown in the figure, then he selects the same from that scrolled window. Later, he decides an intersection fit at that joint. Then, he clicks on the tolerance specification button which pops up the window with the different clearance specification and one with pressfit highlighted (default). Then the designer can select intersection fit from the pup up window and click ok button to indicate the selection to the system. Hence the clearance constraint for the feature-pair changes. The designer can select the functional relationships and constraints depending upon the requirement at any instant of the design process. The designer need not put much effort in calculating the constraints.

The Manufacturing processes of components vary depending on different functions the component is designed to perform. Consider the case of solid cylinder. When the solid cylinder is expected to perform like a pin in a pin-joint, the process for manufacturing is different compared to a shaft in a bearing. These depend on the tolerance limits that are required in the assembly of the components.

One of the basic problems in dealing with geometric modeling is the graphical interaction between the user and the geometry. When the designer wants to edit an object's dimensions, he needs to tell the system which object requires the changes. Some designing systems use the object's ID to identify the object. So the user has to remember each object's ID. The user may have a hard time when there are many objects in the system. Once the designer knows the ID, he needs to input it from the keyboard. This not a convenient way to make a selection.

In our system, we use a select-operate algorithm to operate on an object to select it. The selection is done by a mouse click close enough to the object, then the object is selected. Otherwise, the objects are unselected. The system is smart enough to tell the difference in the distance between a point to an object and the distance between a point and line. Graphical entities, called handles which are a part of feature model are used as interfaces between the model's geometry and the user. Handles are characteristic geometric elements of features representing points and lines of interest. Point handles are used to represent important points on feature geometry such as vertices of block etc. After an object (a feature or a group of features) is selected, handles corresponding to the object will show up as shown in figure 4.6. Object can be editied using these handles. Handles are several sensitive points generated from the object data. Each of the handles relates with one or more parameters of the object. Different handles are designed for different objects because each object has different data structure. The parameter of the object changes according to the handle's behavior. For example, as shown in the figure 4.6, selection of a block to edit, highlights the handles of it. When the mouse is clicked

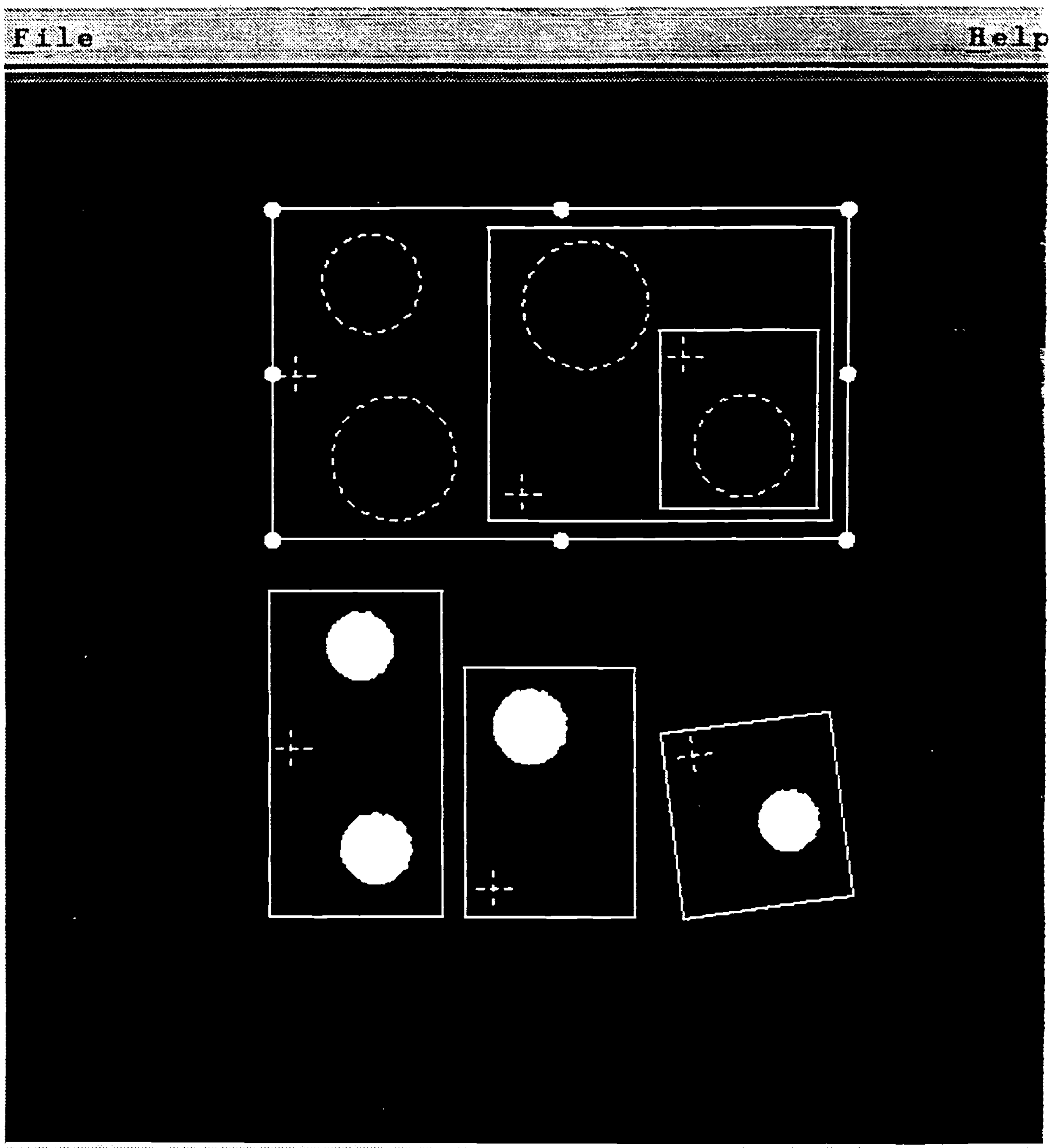

Figure 4.6 Pin Handles used to show the Select-Edit Algorithm

close enough to a handle, the handle will be selected. The selected handle will move with the mouse. The system will interpret the coordinate change of the handle to the parameter change of the object. The object will be updated by the new parameter. New handles will be generated from the updated object. The result is when we drag a handle, it looks like we are dragging the object. This gives the user an opportunity to see the editing procedure animated on the screen. There are 8 handles for a block. Of the eight handles, four are located at the center of four faces and the other four are at the corners of the block. Four handles in the middle of the faces of the block are used to edit the lengths in one direction The other four handles at the corners are used to modify the length of other two adjacent faces simultaneously. If the designer needs to rotate the block, then clicking the right most button of the mouse, pops up a window appears with options of editing and rotating. If the rotate option is selected, then only two handles on two edges appear. Clicking on one of the handles and moving rotates the object.

To modify a feature, the user's intent is to change the parameter of a feature, but still keep its geometry and topology unchanged. And the user may intend to modify a feature-pair by changing the parameters of the feature-pair or the position of it and keep the remaining mating conditions unchanged. As the mouse is the only input device, correct interpretation for each mouse action has to be done. In modifying a feature-pair, there are several possible interpretations for each mouse action. But in all situations, the mating condition of the feature-pair must be maintained.

Handles are used in the select-edit method for editing feature-pairs. When a feature in a feature-pair is selected, the other feature in the feature-pair will be selected. When the parameters of a feature are edited, corresponding parameters of the mating feature are changed to maintain the mating condition of the feature-pair. After strain analysis, the designer might need to change the radius of the pin. If the change in the radius of the hole is not performed accordingly, then the mating restraint will be violated. The pin will be larger than the hole. The interference between parts corresponding to the features is not expected in the designed system. In a pin-slot feature-pair, we also need to take care of the interference problem. The diameter of the pin should be the same as the width of the slot. And the length of the slot should not be smaller than the width of it.

The reference datum has to be specified in the process of assembly and manufacturing of components. The parts of the component of the assembly are positioned with respect to the datum. So when there is a relationship between the components in an assembly, the system checks for the alignment of the mating features. With the data structure we have developed we can check for the relative positioning of the components within the group as well as compare with positioning of mating components in an assembly. This will be helpful in automatic assembly of the components.

Functioning of the System

We will demonstrate how the system can be used to develop a mechanical system. Consider the assembly shown in figure 4.7.

The datums for each of the components is fixed as shown in figure 4.7. The feature-pairs are constructed using the catalog. For example, to construct a feature-pair of pin-in-slot, select the pin as one feature from the feature list and from another list select the slot as another feature. Then the functional relationships that exist between pin and slot appear in the scrolled window under the list of feature. Select the relationship required for the design. The functional relationship is selected as pin-in-slot with press-fit.

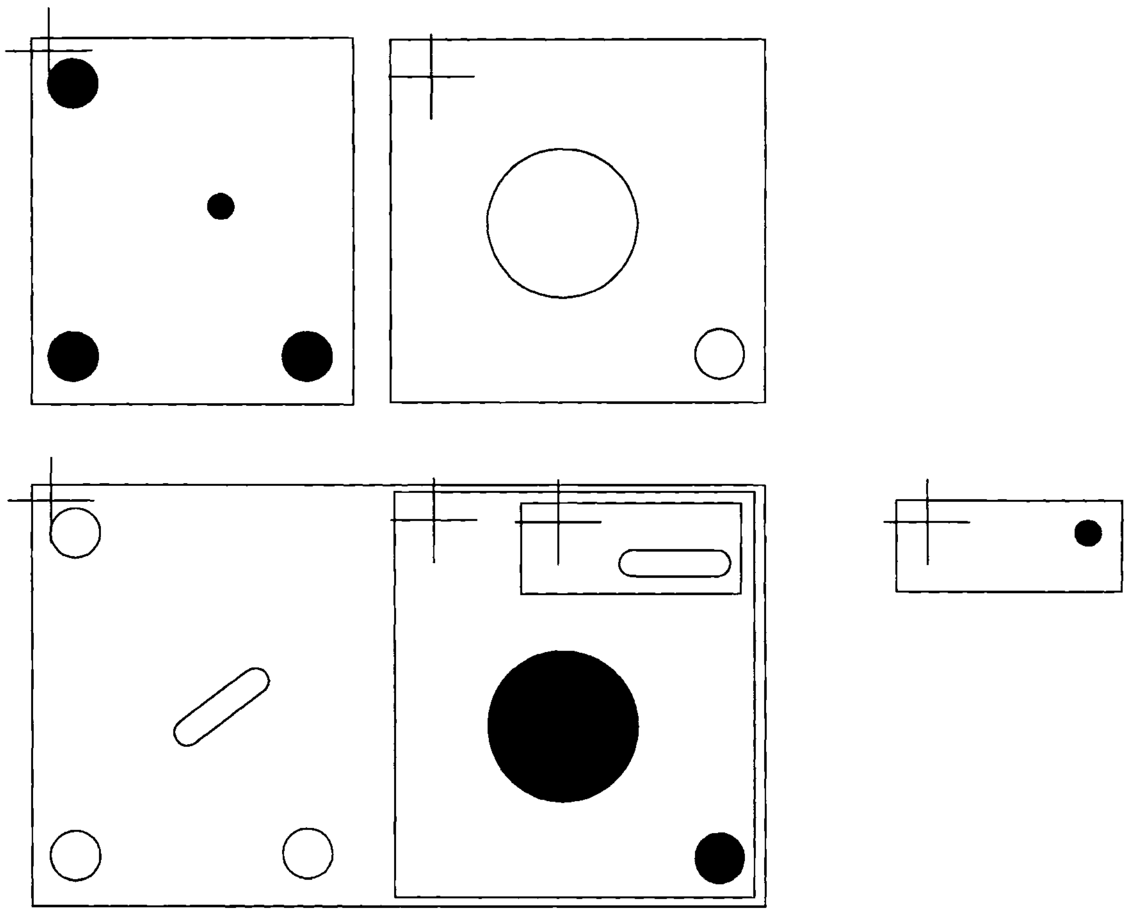

Figure 4.7 Different Components before assembly

Clearance specification can still be edited depending upon the requirement if the defined values are not satisfactory. In this way, the feature-pairs that are required for the assembly are constructed.

The feature-pairs can be separated, keeping the functional relationships between them intact. So, an option is provided for disjoining the feature-pairs, through which features can be moved independent of the mating feature. But still the feature-pair holds all other mating relationships of feature-pairs with respect to their functioning. The features are moved and are attached to different blocks. The features are attached to the blocks. The designer can select the block, then press shift key and select the pin for multiple selections. After selecting all the features to be grouped, then clicking on the grouping icon in the editing tools window, the selected features are grouped. The reference datum has to be fixed for each component if the position tolerances had to be maintained. So, the datums are fixed along the lines of symmetry. A pair of axes are provided as datum. The parts are measured with respect to this datum. The reference datum is provided with handles to vary the size of axes as well as rotate the reference datum to any desired angle. Position of parts on the components are dimensioned with respect to the datum. Tolerances are automatically generated by the system and are assigned to the features. The related features of the feature-pair can be seen by using the viewing option. A red dotted-line appears indicating the mating features of the feature-pairs in the assembly. To check whether the size and position of different features in the assembly are correct, the designer has to choose the assembly option. On selecting this option, the system checks for all the relationships in the assemblies and pops up an error message if any of the parts are misaligned. The system also has the facility for automatically aligning the parts of the components. The designer needs to specify the components that are gaged. The designer can select the group that can be taken as reference and then click on the gage button of the assembly toolbox. So for all checking of the positioning of parts and automatic assembly, the gaged group is taken as reference and any misalignments in other groups are checked with respect to the gaged group. The system makes the gaged component as reference and aligns the other groups of components in proper position with repect to the gauged component.

CHAPTER V

Conclusions

In this system, we use the features to capture the designer's intent for topology and geometry of any single component. The feature-pair concept that we proposed and implemented successfully captures the designer's intent about the assembly at the design phase. The system generates the assembly hierarchy automatically at the design phase and allows the user to alter the designs at every stage of the design process.

A mechanism is developed to represent the different functional relationships that exist between the feature-pairs. The features are catalogued depending on their functional relationship. This functional relationship between the features of feature-pair helps the designer to incorporate certain constraints early in the designing phase. A tree structure has been developed to represent the assemblies of mechanical systems. There are different applications of the structure like automatic tolerances, automatic assembly, dynamic simulation etc. But in this thesis, tolerances and automatic fitting of different components depending on the mating conditions is done. If there is misalignment in positioning of one of the mating parts, then the system can detect the misalignment and then realign their positions so that mating conditions are satisfied. The tolerances and fits that are suitable for the assembly of different components of different assemblies are calculated by the system. The designer is provided with option of selecting the required constraints i.e., either a press fit, loose fit etc. between the components of the feature-pair.

The system has some drawbacks. Because the constraints of the features are hard coded, the user cannot define the new constraints for the feature-pairs. Although the user is allowed to define new feature-pairs using the features, the constraints of these new feature-pairs cannot be defined by the user.

Since this system is developed for designing planar mechanical systems, there are certain limitations in cataloging of the feature-pairs and all the existing features couldn't be represented successfully. Moreover, as the system can represent in 2D and not in 3D, there is some ambiguity in representing mechanical systems using this feature-pair based design system.

Recommendations

In our system, only a few features and feature pairs are provided to the user and those are not enough to represent a complex mechanical system. Although we provide the option of defining the features and feature-pairs, but still all the required features to completely define one really big assembly couldn't be done.

Our system is designed for a 2-D problem. We need to combine with a solid modeller so that it can handle a 3D problem. The process of manufacturing of components depending on the functional relationships between features needs to be improved further. Cost analysis could also be incorporated in the system depending upon the different manufacturing processes etc.

REFERENCES

Cai, P. (1993), A User Interface for Feature-Pair based Design and Analysis of Mechanical Assemblies. Oklahoma State University Master Thesis

Cunningham, J. J., and Dixon, J. R (1988)., "Designing with features: The Origin of Features", Computers in Engineering, Transactions of ASME, 1, pp. 237-243.

DeFazio, T. L., Edsall, A. C., Gustavson, J. A., and Hernandez, J. A (1990)., "A prototype of Feature-based Design for Assembly" , Advances in Design Automation, 23(1), pp.9-16.

Fleming, A. (1988), " Geometric Relationships between Toleranced Features". Artificial Intelligence, v.37. pp.403-412.

Gardan, Y., and Minich, J. (1993), " Feature-based models for CAD/CAM and their limits". Computers in Industry. 23, pp.3-13.

Gossard, D. C., Zuffante, R. P., and Sakurai, H. (1988), "Representing Dimensions, Tolerances and Features in MCAE systems", IEEE Computers Graphics and Applications. v. 8. pp.51-59.

Guilford, J., and Turner, J. (1993), " Representational primitives for Geometric Tolerancing" , Computer-Aided Design., vol 25 no 9, pp. 577-586.

Henderson, M. R., and Chang, G. J. (1988), " FRAPP : Automated Feature Recognition and Process Planning from Solid Model Data", Computers in Engineering, 1988--Proceedings, pp. 529-536.

Hirschitick, J. K., Gossard, D.C. (1986). " Geometric Reasoning for Design Advisory Systems" , Proceedings of the 1986 Computers in Engineering Conference, pp. 263-270.

Hummel, K., Brooks, S (1986). "Symbolic Representation of Manufacturing Features for an Automated Process Planning System", Proceedings of Winter Annual Meetings of the ASME. pp. 233-243.

Irani, R.K., Saxena, M., and Finnigan, P.M. (1990), "Boundary-based Feature Modelling Utility" , Computers in Engineering 1990 - Proceedings. pp. 45-51.

Kang, S. T., and Nnaji, B . O. (1994), " Feature Representation and Classification for Automated Process Planning System" . Journal of Manufacturing Systems., v.12. pp.133-145.

Kim, Y. S. (1992), "Recognition of Form Features using Convex decomposition", Computer-Aided Design, vol 25 no 9, pp.461-476.

Ko, Heedog and Kunwoo Lee, (1987). "Automatic Assemblying Procedure Generation from Mating Condition, Computer-Aided Design, 19(1), pp. 3-9.

Kumar, B., Anand, D. K., and Kirk, J. A. (1988), " Knowledge Representation Scheme for an Intelligent Feature Extractor". Computers in Engineering 1988 -- Proceedings, pp. 543-550.

Laakko, T., and Mantyla, M. (1993), " Feature Modelling by incrementing Feature recognition ", Computer-Aided Design, vol 25 no 8, pp.479-492.

Lee, K., and Gossard, G.C (1985). " A hierarchical Data Structure for Representing Assemblies : Part--I" . Computer-Aided Design, pp. 15-19.

Luby, S. C., Dixon, J R., and Simmons, M. K. (1986), "Creating and Using a Features Data Base", Computers in Mechanical Engineering, vol 5, no. 3, pp. 25-33.

Mantyla, M.(1989), "Directions for Research in Product Modeling, in Computer Applications in Production Engineering", IFIP.

Pratt, M. J. (1988), "Synthesis of an Optimal Approach to Form Feature Modelling". Computers in Engineering 1988 - Proceedings. pp.263-273.

Ranyak, P. S., and Fridshal, R. (1988), "Features for Toleranceing a Solid Model" , Computers in Engineering 1988 - Proceedings. pp. 275-280.

Requicha, A. A. G., and Chan, S. C. (1986), " Representation of Geometric Features, Tolerances, and attributes in Solid Modellers based on CSG" , IEEE Journal of Robotics and Automation, vol RA-2, no 3.

Requicha, A. A. G. (1983), " Toward a theory of Geometric Tolerancing", The International Journal of Robotics Research., vol 2, no 4.

Rimscha, M. V. (1990), " Feature Modeling and Assembly Modeling Approach - A Unified Approach," Advanced Geometric Modeling for Engineering Applications, pp. 203-13.

Salomons, O.W., van Houten, F.J.A.M., and Kals, H. J. J. (1993), " Review of Reseach in Feature-based Design" , Journal of Manufacturing Systems. v. 12 no. 3, pp. 113-132.

Schulte,M., Weber, C., and Stark, R. (1993), " Functional features for design in Mechanical Engineering". Computers in Industry vol 23, pp. 15-24.

Shah, J. J., Rogers, M. T. (1988), "Feature-based Modelling Shell : Design and Implementation", Computers in Engineering 1988 - Proceedings. pp. 255-261.

Shah, J. J., and Rogers, M. T. (1988), "Expert form feature Modelling Shell", Computer-Aided Design, 20(9), pp. 515-524.

Shah, J.J., and Bhatnagar, A. S., (1989). "Group Technology Classification from Feature-based Geometric Models", Manufacturing Reviews, 2(3), pp.204-213.

Shah, J. J., Ravi, T. (1992), "Feature-based Assembly Modelling", Computers in Engineering 1992: Proceedings of the 1992, ASME International Computers in Engineering Conference and Exposition, pp. 253-260.

Shah, J. J. (1991), "Assessment of the Features Technology". Computer-Aided Design, 23(5), pp. 331-341.

Shah, J. J., and Mathew, A. (1991), "Experimental Investigation of the STEP Form-Feature Information Model" , Computer-Aided Design, vol 23 no 4. pp.282-296.

Sheu, Liang-Chyau., and Lin, James. T. (1993), " Representation scheme for defining and operating form features" , Computer-Aided Design., vol 25, no 6. pp.333-347.

Stewart, N. F. (1993), " Sufficient condition for correct topological form in tolerance specification" , Computer-Aided Design. vol 25, no 1, pp. 39-48.

Tikerpuu, J., and Ullman, D.G.(1988), " General feature-based frame representation for describing mechanical engineering design developed from empirical data".

Turner, G. P., and Anderson, D. C.(1988), "An Object-Oriented Approach to Interactive Feature-based Design for Quick Turnaround Manufacturing", Computers in Engineering 1988 - Proceedings. pp. 551-555.

Unger, M. B., and Ray, S. R. (1988), "Feature-based Process Planning in the AMRF', Computers in Engineering 1988 - Proceedings, pp.563-569.

Roy,U., and Liu. C. R. (1988), " Feature-based Representational scheme of a solid modeller for providing dimensioning and tolerancing information", Robotics and Computer Integrated Manufacturing, vol 4 no 3/4 pp.335-345.

Vaghul, M., Dizon, J. R., Sinsmeister, G. E. (1985), "Expert Systems in a CAD Environment: Injection Molding Part Design as an Example", Proceedings of the 1985 ASME Computers in Engineering Conference, Boston, MA.

Wang, N., and Ozsoy, T. M. (1991), " A scheme to Represent Features, Dimensions and Tolerances in Geometric Modelling", Journal of Manufacturing Systems. v. 10 no 3, pp. 233-240.

Wehage, R. A., and Haug, E. J. (1982), "Dynamic Analysis of Mechanical Systems with Intermittent motion". Transactions of ASME. Vol. 104.

VITA

Sudheer Gaddam

Candidate for the Degree of

Master of Science

Thesis: FEATURE PAIR BASED DESIGN: DEFINING AND APPLYING FUNCTIONAL RELATIONSHIPS BETWEEN COMPONENTS IN ASSEMBLIES

Major Field: Mechanical Engineering

Biographical:

Personal Data: Born in Hyderabad, A. P, India, July 19, 1972, the son of Vimala and Ram Reddy Gaddam.

Educational: Graduated from St. Marys Junior College, Hyderabad, A. P, India, in May 1989; received Bachelor of Engineering Degree in Mechanical Engineering from Chaitanya Bharati Institute of Technology, Hyderabad, A. P, India in July 1993, completed requirements for the Master of Science degree with a major in Mechanical Engineering at Oklahoma State University in May 1995.

Experience: CAD/CAM consultant in CADVISION Engineers Pvt, Hyderabad, India, August 1991 to July 1993. Practical Training at Hindustan Machine Tools, Hyderabad, India, 1993. Employed as a Research assistant, School of Mechanical Engineering, Oklahoma State University, May 1994 to August 1994. Employed by College of Engineering, Architechture and Technology as a Research Assistant, January 1994 to December 1994.