GRAPHSUPPORT: Interactive Modelling with Computer Graphics in Assisting Design

Ö. KARAÇALI

University of Technology, Loughborough, England
E–Mail: O.Karacali@lut.ac.uk.

Synonym

This paper explains a novel approach that the designer can be supported by interactive objects modifications with graphics system. An engineering application, called ***GraphSupport*** system is identified and relations to the software support aspect is discussed. An example of industrial product in interactive modelling with objects is applied to ***GraphSupport*** to aid engineering design. Simple, yet versatile, information and functional modelling methods are introduced, allowing variants and views of the product to be represented and modified in a solid modelling environment.

1 Introduction

Currently one of the critical problems for solid modelling applications with **computer graphics** in CAD/CAM(Computer Aided Design/Manufacture) integration is product oriented data modelling, human interaction and the efficiency of using the information in design and manufacture. Various approaches to solid modelling with computer graphics have been taken to solve the problem. Much research concentrated on CAE/CAD with computer graphics improvements. Including:**Requicha** *[1]* summarised historical view of the Solid Modelling with computer graphics research directions and as a result of this, commercial developed graphics system in design and manufacturing applications. **Pratt** *[2]* outlined the last three decades development on techniques for automatic recognition of features, design by features, and exchange data between solid modelling systems. **Sabin** *[3]* identified numerical definitions for computer graphics to interact with design, and to answer "what does the shape look like?","what shape must templates be for manufacture or for mating structural members?" questions. **Teixeria** *[4]* explained novel integration methods of constructive geometry and technical design knowledge, in the dialogue with geometric modelling with computer graphics systems. **Encarnacao** *[5]* extensively reviewed 1990s graphics technology, outlined trends and integration methods with graphics in application. **STEP**(Standard for Exchange of Product Model Data) *[6]* community explores the data model for a structured source of all product related data to ensure the product data integrity and searches interative graphics system such as **Parasolid** to apply their problems for testing.Using low cost workstations with outstanding graphics capabilities, the application engineers and research scientists mentioned above have produced methodologies and basic software techniques interacting with capabilities which are at a level of refinement that is required for subsequent development of design and manufacturing applications. Although current approaches are impressive for their information storage and computational capability, they are still not capable of solving many problems in creating frameworks of cooperative environments of industrial engineering application. Visual display of the product is one of the most important issues to give a perspective to the user as well as providing structured information.

A computer based process should take into account the questioning nature and the uncertainty of the design activity. Design is seen to be not simply about the production of images of an object, be they in pictorial form or as an engineering drawing. It is more about finding answers to engineering questions. It is more important to the designer to have high quality data than simply graphics. These answers may however be provided as pictorial information, as in a geometric representation.

The ideal CAD/CAM system should be one in which the input instructions are not simply those to initialise graphics generation or display functions, but one in which the input request is for engineering information of relationships of entities, functionalities etc. The output, in terms of graphics, should thus be seen as a means of providing those answers. Such a system should be hierarchically structured to align with the interactive nature of the engineering approach. The system should also be structured to allow the problem to be contained with minimum data. The advance enhancements of CAD/CAM in computer technology has been driven by the development of greater computing power and graphics capability. The question in design for the manufacturing environment is how to provide reliable information to the designer by using a graphics system. Graphics is one of the tools to aid the designer, but is it only visual display? Should further effort be put into the development of ever more complex modelling and graphics systems. What are the benefits for the designer? In answering these queries, one needs to analyse the role of both the design process and CAD/CAM.

The aim of the research is to develop a systematic approach to converting existing intermediate workpiece specifications and drawings and associated information from being paper based to a digital format. To achieve these goals the research needs to investigate real users working with CAE/CAD/CAM systems. A methodology is being developed to organise design information generated by these systems having regard to the retrieving of this information for reuse in future product designs.

The balance of the paper will discuss as follows: a model oriented design engineering system with graphics system in the next section, an environment of engineering application expert to support engineering design system in the section two, the latter graphics related to an application is given and information and functional modelling methods are given at the end.

This paper is involved in *"Design"* issues with graphics system so as to aid design. The architecture of such a *Design System software* is currently being applied to develop an information system to exploit information models for the experimental machined components presentation on a graphics system and software. The following section outlines the ***GraphSupport*** research project.

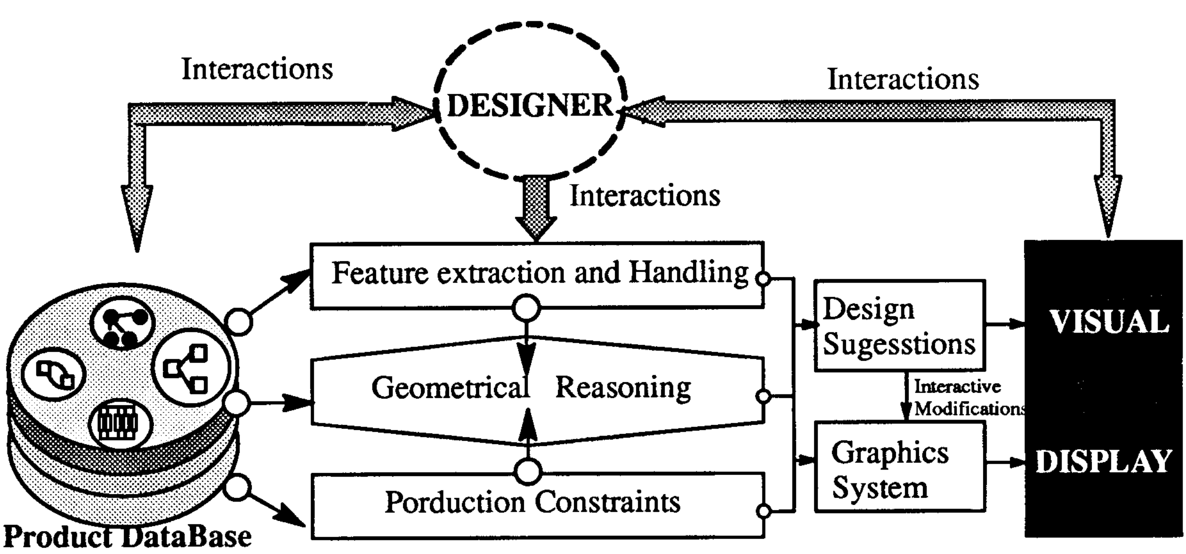
3 The Environment of the Engineering Application to aid designer by *GraphSupport*

How can we use computer graphics systems to support product development concept? Human beings are presently used to implement Design For Manufacture, Assembly rules for development of a product. In order to compete successfully in the world-wide market place, the design environment should be implemented in computerised systems to achieve this goal. The design process is a complex area. The system in design operates on the basis of exploring the engineering relationships of a real problem rather than by following some set procedure. Computer technology enhances the industrial product design development, primarily in manufacturability, assemblability, appearance and human usage. Industrial product design is biased towards the visual while engineers focus on performance. They are often at odds, yet successful product design clearly requires the skills of both. The separation of form and function in the design stage can lead to difficulties in manufacturing products.

The approach within Computer Aiuded Design system seeks to integrate functional design and manufacturability engineering concerns the initial and detailed phase of the product design. We have developed an experimental computer aided design tool that interacts with design and manufacturing engineers in order to implement the subjects mentioned above as shown in Figure–1. The ***GraphSupport*** is an application domain that supports design and manufacturing functions in order to enterprise a plan via graphics system. The ***GraphSupport*** generates reliable product and process data for design and production engineers by using product data model and considering cost–quality–times for manufacturing. This software application generates ***Design Support Information*** and constraints via presentation of a visual display. The *GraphSupport* has a graphics system facility to interact with users on display.

Every design our system generates can be produced using the straight presentation of product. We chose to focus on products such as machined parts, taking the product specifications. Figure–1 illustrates the system developed for providing reliable information to the designer to assist engineering design. With this system, a designer can produce a first initial design and interaction on computer that provides an information support system and interactions between designer and the support environment make product design iterations. Finally, the designer gives the program a desired orientation for product display. In a short time, the system proposes component configuration according to manufacturing rules so the designer can browse through alternatives and make selections for further development, and look at the manufacture information to make decisions. The program uses a product model to store productin formation which can be expanded for new products. Using a product model and graphics, the presentations of new products can be developed quickly and conveniently. This eliminates many of the designer's repetitive tasks. Product details such as basic element features, tolerances are automatically installed in the appropriately detailed design. The designer creates such a detail, and the programs adapts the details to suit specific product requirements. The renderings provide realistic images from which designers and clients can make preliminary decisions.

The designer choses some initial descriptive specifications such as user inputs and accesses for entering data product model. The software program of the ***GraphSupport*** then uses rules to configure the product to suit the chosen components and specifications. We require our system to produce a visual display for the users. This approach is consistent with the way many designers work during the development phase, and it differs from another system called ***"design and throw to manufacturing"***. For example a program using our



Figure–1 System design and interactions to shape modelling environment

approach takes skeletal products from initial design and integrates with machinability concerns. A rule based approach builds in the ***GraphSupport*** an intermediate structure using *product data models*. We chose this architecture because manufacturing information is readily registered for a specific company. The system of the ***GraphSupport*** establishes the product face and region that components will occupy without precisely locating them. Next the system locates the component physically thereby defining the product's volume. A component's class attribute determines which set of rules applies and the detail attribute is used at the detail level to determine what finishing element a component requires. First we use component dimensions in each part to estimate overall product dimensions, and to select the layout that best matches the product attributes of height and use.

The ***GraphSupport*** comprises two functional elements: Feature extract and handling. The feature extract is to recognise features to make decisions. The feature handling Element will determine the objective function to be applied in any given situation and the constraints applied will be added to the technological constraints. In order to recognise these features in the ***GraphSupport***, a ***Solid Modeller [8]*** is used to handle these issues. Features of a product by using a *Solid Modeller*, first by calling functions to generate and then join the features together to be incorporated into *the Knowledge Based System*(KBS). This will be verified by human expertise and decision making processes to create this design automatically. The *Solid Modeller* is being used to visualise and modify the product design in a three dimensional form. Component Specifications–Shapes can be described as an in–progress workpiece that results during the transformation from the initial blank to the finished part. The intermediate specifications associated with each operation are specified at the beginning, i.e. oversize dimensions and tolerances due to fact that there are transformations of geometry as well as dimensional changes. ***Geometric Reasoning*** in the ***GraphSupport*** acts on a knowledge structure to make inferences, that is, to make explicit some of the knowledge implicit in the structure. The geometric reasoning element provides information based on the notion of functional reasoning which has the ability to integrate shape, function and planning in reasoning about the manufacturability of products.

The ***GraphSupport*** recognises the parts in the product and *Geometric Reasoning* which takes place on a 3–D representation of a scene, reasons on the features and associated processes and available resources. A proper model for the shape and function such as manufacturability relation could provide a powerful aid to an intelligent planning system. Functional information can be associated to geometric object, thereby allowing more general object classification and recognition.

A central strategy system coupled with a set manipulation engine should provide a common functionality to support the areas mentioned above together with necessary specialist knowledge as illustrated in Figure–1.

The design suggestions will be introduced to the design environment to display the alternative configuration of part shape modifications, and technological parameters. In the ***GraphSupport*** next section Object–feature information in is discussed.

3 Features Used in Embodying a Part Design in the *GraphSupport*

One of the key research issues in the ***GraphSupport*** is Product data modelling *[7]* that captures the design and functional aspects. The feature data provides constraints on how the feature should be produced in product modelling.

This feature can have geometrical aspect and manufacturing aspect data that has constraints related to the production of a feature. In order to overcome the limitations of operations in CAD, the concept of using form features (Shape Elements) for modelling is being used. A feature in this research is defined as an object. The object concept also includes all physical things in the system, sometimes we will refer to the *feature* as an *object*. The object is a geometric shape defined by a parameter set, that has special meaning to design or manufacturing engineers. Representation of objects will be displayed by Parasolid *[8]*.

The Parasolid package provides geometry and solid modelling functions. The bulk of the information needed in the design and manufacturing deals with the geometric shape of the product. Features represent a collection of entities in a form like hole, groove, threat and thus provide information at conceptual level. The use of such groups of geometry coupled with the necessary information needed for modelling in the ***GraphSupport***.

GraphSupport is important to aid design using manufacturing information to satisfy a feature produced by a machine in this research as shown in figure–2. The designer puts his initial design on visual display.The system offers interactively new alternatives. If it is not accepted, the reason for that is shown on screen. For instance the design for manufacture application outlines that there are problems producing a suitable corner without some modification to the geometry. The designer is informed that there are several possible solutions to the manufacturing of the cylinders and to connect to a shaft object there must be either a tapered, a chamfer, or a round corner as shown in figure–3.

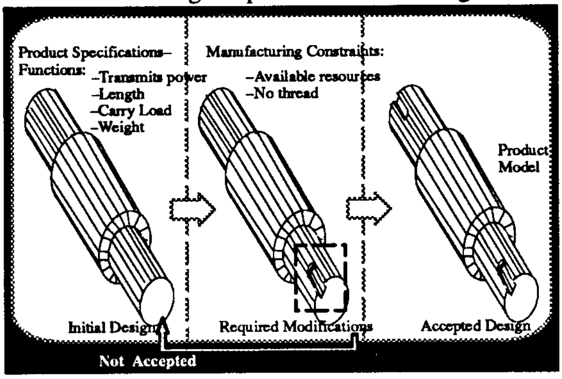


Figure.2: Designer interrogate and modify geometry

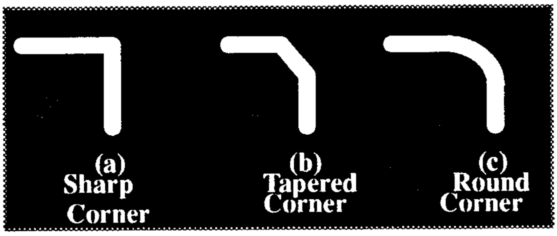


Figure.3: Alternative designing corner shapes

The correct scaling of these corner treatments is essential during the application to the surface of the product. Ideally the industrial designer would interactively choose appropriate scaling functions on a trial–and–error basis during the design process for manufacture. In our research, architecture is based upon a feature based part modelling system. The trends towards capturing design and manufacturing are oriented not only at geometry but also at capturing a product identification and more information included in the model to make use of CAD databases.

The integrated environment will provide a framework to integrating the design and manufacture. Feature based design and feature extraction have an important role in associating design and manufacturing information. The manufacturability information of a part feeds back the accurate and realistic information to the designer. The developments of creating reliable information mentioned in the ***GraphSupport*** interact with the Information Models and engineering knowledge and graphics system. Support of the designer by means of the ***GraphSupport*** has logical functional and data modelling methods that are explained at the end of the paper. In the next section we demonstrate how our approach with Parasolid can aid product design development. We show that the system can create designs both meaningful to humans and distinctive over a range of products. We intended that it be realistic enough to illustrate the models validity.

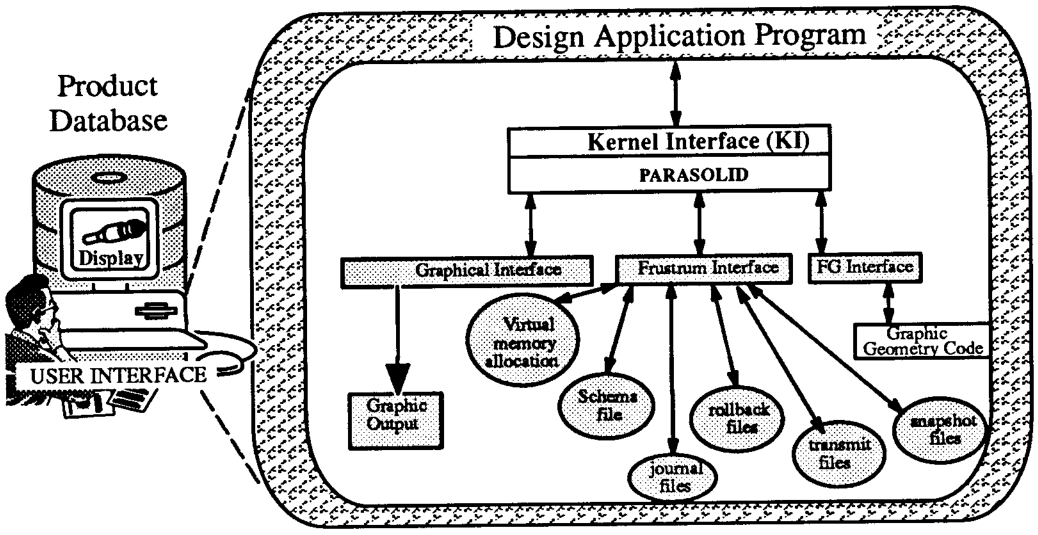


Figure.4: Parasolid concept and Interfaces

5 Parasolid Concept and Connections to Design Environment

This section gives the concept of Parasolid and the environment to interface with the designer. Parasolid is an exact B−rep (Boundary Representation) geometrical solid modeller *[8][9]*. Parasolid*[10]* is a part of the **UNIGRAPHICS.** Parasolid used in this research as a test for the ideas is as follows: build and manipulate solid objects and combine them into assemblies; calculate mass and moments of inertia; output the objects in various ways, including diagram; store the objects in the Objectivity*[11]* database in order to archive and retrieve them later. Parasolid is designed to be centre of the system that is based on 3D model data. It is connected to the product model when it is triggered off, in order to get visual display according to user inputs of product specifications.

As shown in Figure−4 Parasolid has two interfaces to the outside world. One is the Kernel Interface (KI) that is mounted 'on top' of the modeller. KI is used to control the functioning of the modeller, and to program it to model and manipulate objects. The other interface lies 'beneath' the modeller and consists of two parts:Frustrum and Graphical Output. ***The Frustrum*** is a set of routines written by the ***GraphSupport*** system builder. They are called when data needs to be saved or retrieved. The user needs to decide how to manage the storage of data which the kernel outputs through the Frustrum. This involve determining the format and location of the files when writing the Frustrum routines. ***Graphical Output (GO)*** routines are written the same as Frustrum. However, unlike the frustrum, the output from these routines is not data files, but rather instructions to the graphics system for drawing pictures requested from kernel. Parasolid in the ***GraphSupport*** calls GO routines as a by−product of having been asked to draw something by one of the rendering routines of the KI.

The kernel prepares drawing data piecemeal, so that one call to a rendering routine will normally generate many calls to the GO routines. ***Foreign Geometry (FG)*** is where Parasolid reads or writes geometrical data that has been generated by another modeller and that cannot be represented by the structures inherent to Parasolid.

Parasolid in the ***GraphSupport*** stores models in the form of B−Rep, ie it stores a model by their boundaries, although the package also allows geometry to be built up using constructive solid geometry (CSG).The entities in the Parasolid kernel are grouped into three overall classes. Their names

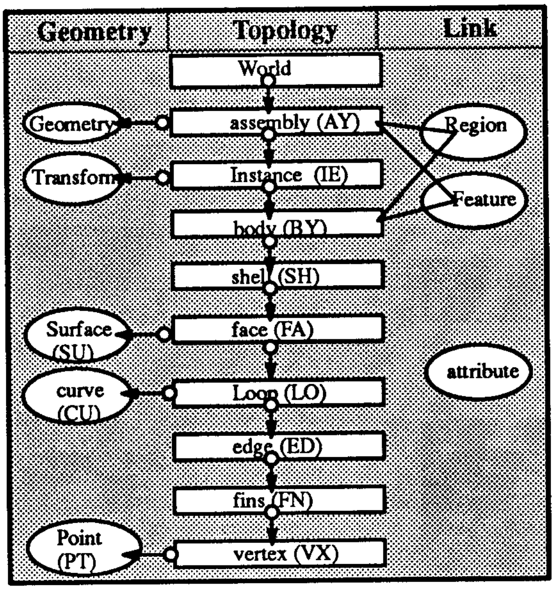


Figure.5: The entities of the Kernel Model

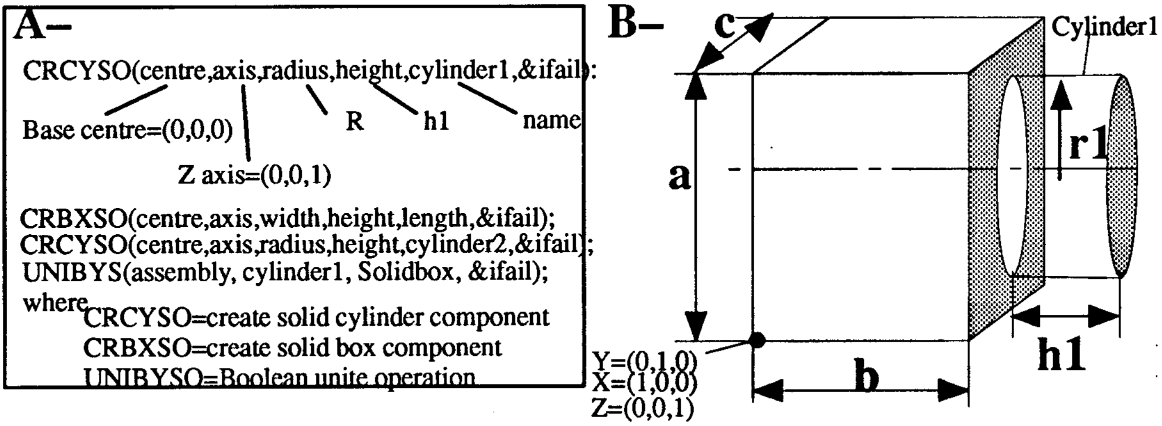
and relationships are shown in Figure−5. The *geometric entities* specify the geometric relationships within a body or an assembly. The *topology* contains all the entities which constitute the structure or skeleton of the model. *Associated data entities* allow aadditional data to be manipulated and attached to the model, or extra structure to be defined. The program in the ***GraphSupport*** with Parasolid is written in C programming language which calls the KI routines as part of the modelling system to start and stop the modeller on SPARC station IPX. .

From Figure−5 geometry can be attached to the topological hierarchy. Edges and faces have no geometry, and therefore complex surfaces can be represented by attaching surfaces to faces and curves to edges. The next section describes how a feature is created in Parasolid and the application for interactive product modifications in the ***GraphSupport***.

6 Interactive modelling with objects in Parasolid to Support Design.

In this section an example of design with graphics is given and the creation of functional design and machinability considerations in interactive product model modifications is explained. In Figure−6 a representation of objects is shown. The Figure−6 illustrates the generation of information for the functional cylinder, which is equivalent to a shaft, ie CRCYSO, create cylinder, CRBXSO, create a box solid, UNIBYSO, unite the cylinder1 into solid box. Two type of information are given as a mathematically defined entity for the creation of the object. Mathematical presentation of the feature provides a computational view for the part analysis whereas visual display gives a explicit view for the reasoning and recognition. The use of feature presentation for initial product data and interactive product modifications facilitates the design operation. The representation of the new product should be easy to comprehend in terms of the geometry and feature types so that the design and manufacturing processes can be reasoned. Geometry information should include feature information for ease of analysis. For interactive interfacing with the models, the representation of the product should be given in different alternatives. Each feature information should be associate with strategist and the ***GraphSupport*** for manufacturing constraints from the manufacturing model. The reasoning process in the ***GraphSupport*** aids the definition of features. The creation of a product definition of a product such as a shaft is only given for geometry information consideration, although the product definition also includes tolerance and materials information etc. The ***GraphSupport*** requests that the designer specifies the diameter of the shaft, the volume and the envisaged loading as functional constraint information to embed into the Product model. The diameter specification gives the dimensional perspectives to compare existing objects. The volume specification provides the height of the same cylinder. The loading is used to calculate the required height and therefore provides the height and base coordinates of the cylinder.

The designer having chosen to create a part of the shaft as shown in Figure−7, the ***GraphSupport*** would then interrogate the Product model in order to discover the meanings in terms of geometry, the parameters required for the product type, and therefore the information to be requested from the designer. The information in the Product model should consist of structured processing information and instructions on how to generate the information for the individual product required by the designer. In order to generate the individual product generation information, the ***GraphSupport*** will be required to provide the data within the parentheses, ie the parameters for the generation routine. The ***GraphSupport*** is, in this way, able to generate the information for the functional geometry shown in Figure−7A. To generate part of a shaft one command is required, CRCYSO, create a surface tolerance attribute and attach it to the surface, SETLEN, fix (associate) the appropriate to the edge which has the tolerance attribute. The geometry creation information is therefore modified to include these routines.

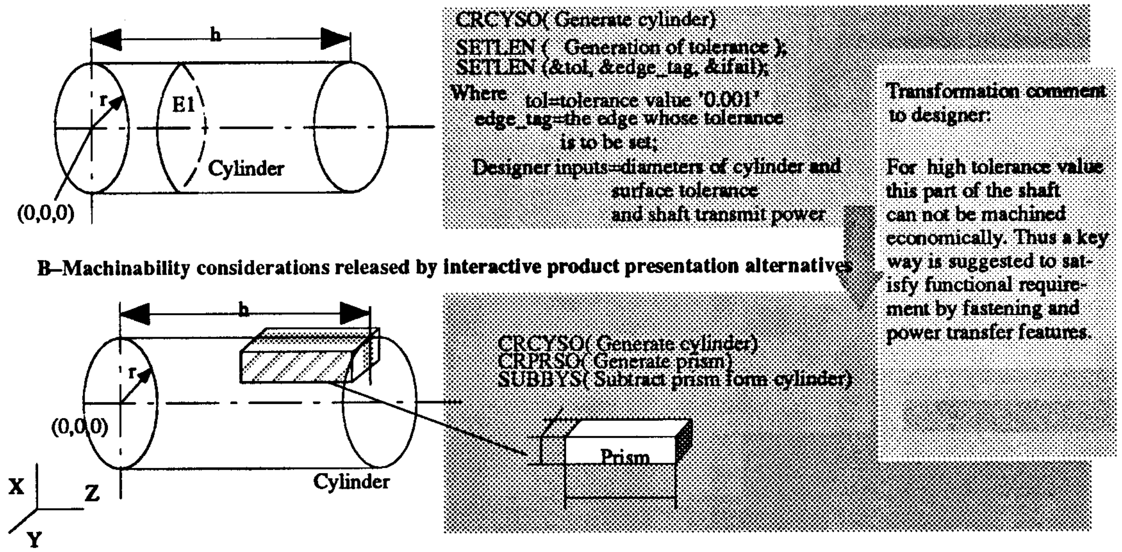


```
CRCYSO(centre,axis,radius,height,cylinder1,&ifail):
CRBXSO(centre,axis,width,height,length,&ifail);
CRCYSO(centre,axis,radius,height,cylinder2,&ifail);
UNIBYS(assembly, cylinder1, Solidbox, &ifail);
where CRCYSO=create solid cylinder component
      CRBXSO=create solid box component
      UNIBYSO=Boolean unite operation
```

Figure.6: Object assembling generation by using explicit (B) and implicit (A) features

A– Creating Functional Feature by designer

B–Machinability considerations released by interactive product presentation alternatives



```
CRCYSO( Generate cylinder)
SETLEN (  Generation of tolerance );
SETLEN (&tol, &edge_tag, &ifail);
Where  tol=tolerance value '0.001'
   edge_tag=the edge whose tolerance
          is to be set;
  Designer inputs=diameters of cylinder and
                  surface tolerance
                  and shaft transmit power
```

Transformation comment to designer:

For high tolerance value this part of the shaft can not be machined economically. Thus a key way is suggested to satisfy functional requirement by fastening and power transfer features.

```
CRCYSO( Generate cylinder)
CRPRSO( Generate prism)
SUBBYS( Subtract prism form cylinder)
```

C– Generation of an object by extracting parameters in the *GraphSupport*

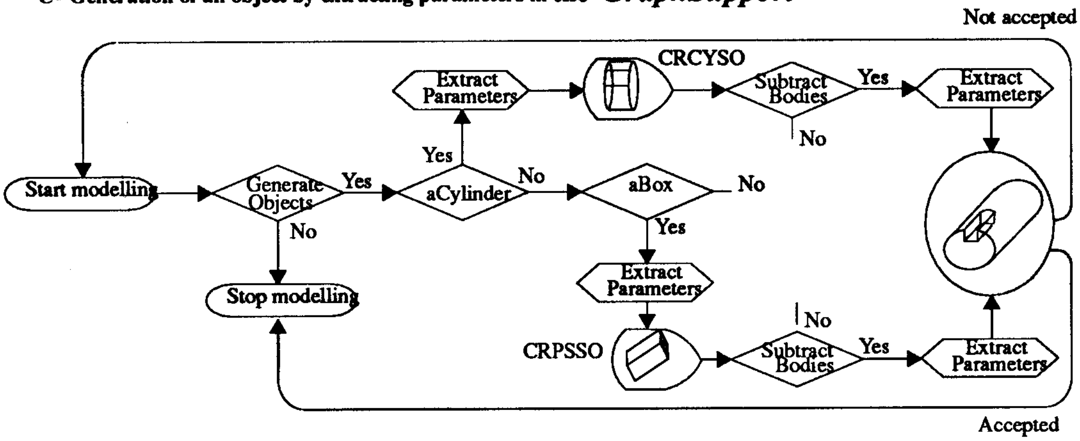
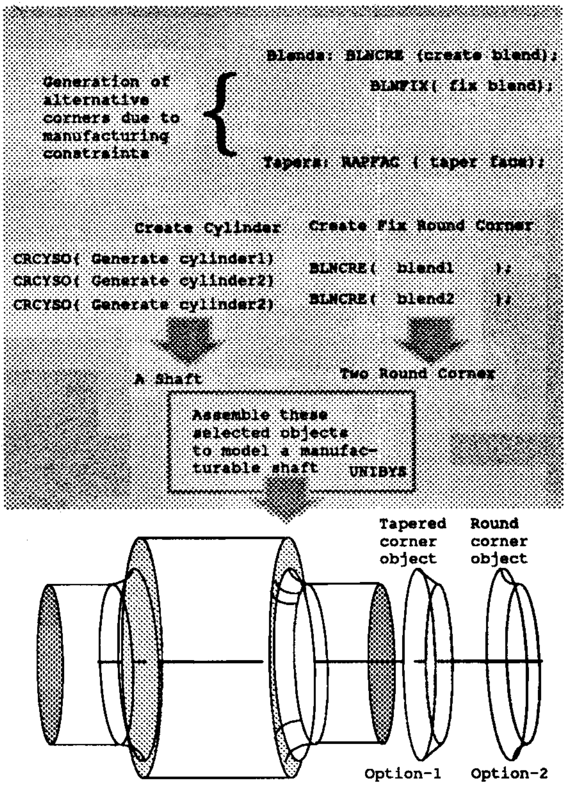


Figure.7: The processes in the modelling part of a shaft in the *GraphSupport*

The above creation information for the functional geometry must then be modified by the the *GraphSupport* in order to provide manufacturability. Thus the manufacturability constraints in the Manufacturing data in product model must be applied. Figure–7B shows the application for creating of a key–slot for machinability. As a result of the lack of machine tools for high quality feature machining e.g expensive tolerancing to produce this surface quality the designer is advised to create a key way. Since the feature type e.g. functionality feature etc is known it is easier to decide on the manufacturability due to manufacturing constraints. The application of the manufacturing constraints will also enable the *GraphSupport* to calculate the parameters to be contained within the parentheses of the new routines, for example the tolerance attribute will be turned into an additional key way feature that will come from the constraint information. Figure–7C gives the flow diagram for generation of an object by extracting constraints (called parameters) to aid designer decisions. The *Extract parameters* command in Figure–7C implies that it is the information received from the user and extracted from information models. Interactive product modifications in the *GraphSupport* is verified by creating the features and alternatives via information models, users, visual displays. Figure–8 illustrates the addition of a round corner to a shaft by the designer. By creating cylinders, a shaft is first roughly generated. Cylinder primitives are united to create a complete shaft that includes Cylinder1, Cylinder2, Cylinder3. Using envisaged feature representation, a blended corner is generated. To join the round corner the shaft UNIBYS (Boolean unite operation) com



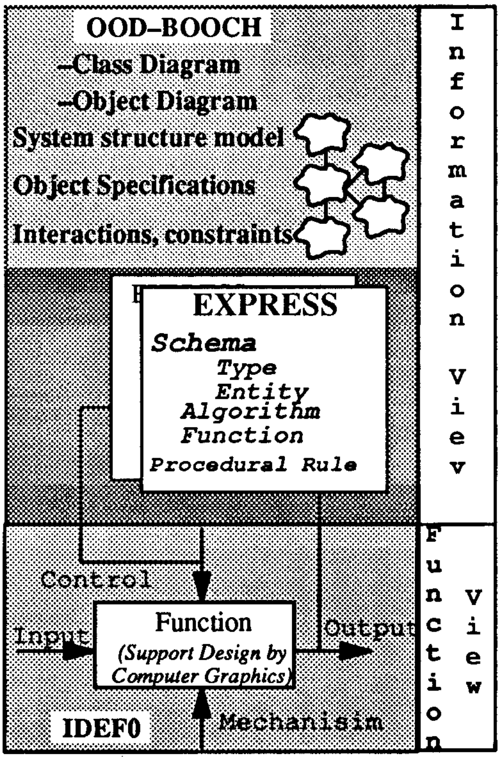
Figure–8 Product modifications by interactive modelling objects

mand is used. The blend is treated as a round corner with respect to the constraints that will be extracted from the manufacturing model to facilitate a manufacturability analysis. This application proves that interactive modelling with objects can be successfully managed in the *GraphSupport* to support engineering design.The next section provides the functional and information modelling techniques to support graphical design activities in the computer.

7 Modelling Techniques and Tools Used to Support Design with Computer Graphics

It is very important to have a good understanding of the problem that leads to designing models that emphasize the proper and effective structuring of a software system as well as defining the relation and interactions between the system's models. The designed models help to reason about the system's structure and provide a requirement to implementation. The information and functional modelling implementation in the *GraphSupport* is required to develop a framework to support engineering design. While functional modelling IDEF0 is used to show the support activities for design with computer graphics, Object–Oriented Design Booch methodology and EXPRESS is employed to describe the relationships between Information models and engineering application. Figure–9 illustrates graphically the brief description of the informational and functional modelling techniques used in this research.

Functional modelling has been done by IDEF0 **[13]** as a basis of the *GraphSupport* functional view. In this research IDEF0 provides a graphical representation and a hierarchical structure to define activities that can be executed simultaneously, or serially. In Figure–9 an IDEF0 in abstract level is illustrated. The boxes are used to represent functions, and arrows represent flow of information. A function has related flows into inputs, outputs, controls and mechanisms. In IDEF0 inputs are transformed into outputs under controls using mechanisms. IDEF0 has capacity to incorporate constraints and supplies crucial base to show activity interactions and a clear structure to explore design interactions with product data model that is suitable for engineering design. In Figure–10 the IDEF0 activity modelling is applied to a design analysis framework in the *GraphSupport*. *"Analyse feature"* allows identifying the feature objects in product model and related geometry information, whereas *"Extract constraints"* provides which constraints should be extracted from the manufacture model. *"Extract constraints"* analyse the information of features to interpret and advise manufacturability information to the designer. *"Analyser part"* examines the product and product related manufacturing information by interacting with *"Analyse feature"* and *"Extract constraints"*. *"Analyser part"* transforms product specifications into geometry information. *"Implement solid model"* is the environment for product modifications and verifications to be done. Figure–10 illustrates the acceptable interactions in the subtasks to make a design analysis.



Figure–9 IDEF0, BOOCH and EXPRESS descriptions to support engineering design

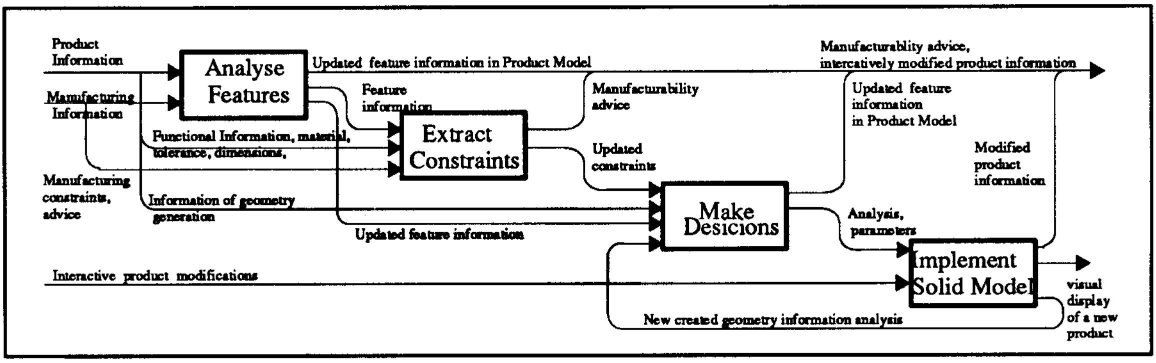


Figure.10: IDEF0 Functional modelling of Design analysis by the *GraphSupport*

Booch method [12] is defined as a method of design encompassing the process of object oriented decomposition and a notation for depicting both logical and physical as well as static and dynamic models of the system under design. The ***GraphSupport*** system is being developed in the Object–Oriented Design (OOD) by the Booch Methodology. The implementation has therefore been done in Rational Rose Graphics–UNIX SUN SPARC Station. The reason, applying this method for the ***GraphSupport*** is to represent Product and Manufacturing Information and to establish detailed relations at different levels of abstraction and to aid the designer. A class diagram shows the existing classes in the system, however an object diagram provides object information. In Figure–11 an object diagram is illustrated to show constraints and relations between objects for *"aShaftDesign"*. The objects communicate by passing messages to each other. A scenario is built to show the relations between *"aShaftDesign"* object and "aCylinder, aBlindSlot, aMetalMaterial, aCutter, aFixture, aMachiningCenter". *"aShaftDesign"* object, for a given shaft product specification which controls the available objects to see whether they are satisfying the requirements and achievable targets. *"aCylinder"* object is controlled against machinability, relation to other objects and available manufacturing resource and capabilities. The Dimension value of the *"aCorner"* object is compared to *"aCylinder"* and the machinability constraints object to check dimension values.When the *"aCylinder"* object receives messages from *"aCorner"*, The ***GraphSupport*** database manager will tell the *"aCylinder"* object to create relationship context details. Messages will sent to other objects to generate a list of the attributes and parameters according to attributes. The Database manager retrieves needed information and invokes a message to display the geometrics associated with the ***GraphSupport***. The scenarios drawn in this diagram are useful to give feedback information to the designer. The product model is modified during the checkings against manufacturing constraints. The designer is introduced to additional object information while observing the objects on graphics visual display.

As a complementary data description EXPRESS language is being used to enable the ***GraphSupport*** data representation to define the constraints im– posed on objects, and defining the operations.

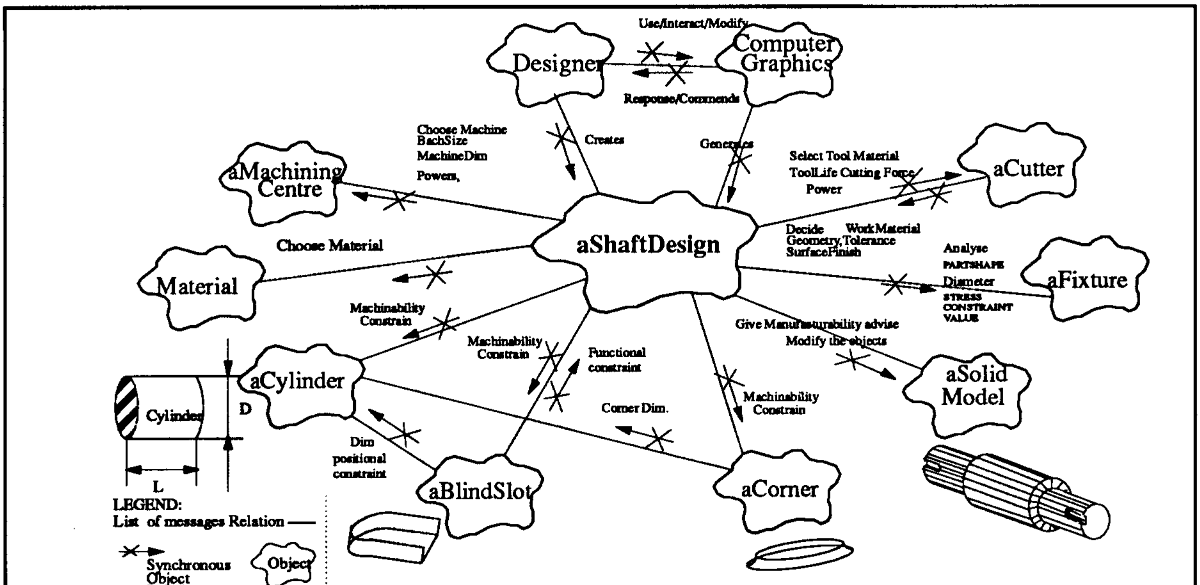


Figure.11: The Shaft Design Object Diagram

EXPRESS Information modelling [14] language defines constraints as well as data structures. For this purpose we chose the data modelling language designed as part of the STEP international standard. These constraints describe a correctness standard that must be met before an engineering data set is sent to others. The main descriptive elements of EXPRESS are illustrated in Figure–9. The EXPRESS supports the engineering design requirements in the ***GraphSupport***, that allows: sharing data by defining common EXPRESS schema for that data, an application software written in Object–Oriented Programming language(e.g. C++) to share data, concurrent changes need to be merged using Object–Oriented data files.

8 Summary

In this paper we have presented an information support system architecture under development. An engineering application called the ***GraphSupport*** is discussed to support designer with graphics. We have shown that interactive modelling with objects can be realized in the ***GraphSupport***. The interaction of data from the product database is used to provide feedback to the designer as the product definition is developed. Our research aims to develop tools to help design. Our system creates simple features on the desktop. the ***GraphSupport*** system produce meaningful configurations according to manufacturability considerations. It can create multiple design alternatives with recognisable identities. With realistic computer graphics, the designer can quickly evaluate and select from designs that the system proposes. These designs serve as the basis for further development, and their renderings can conveniently communicate ideas for that purpose. Providing a link of the system to visual display in integration environment with the ***GraphSupport*** would allow interactive object modelling realization of design with Parasolid. Further we have presented information and functional modelling techniques (BOOCH/IDEF0/EXPRESS) to support the designer with graphical representation.

Acknowledgments

The work presented here has been financially supported by the Ministry of Education, Ankara–TURKEY.

9 Bibliography

[1] Requicha A.A.G, Voelcker H.B. 1983"Solid Modelling: Current Status and Research Directions" Journal of IEEE Computer Graphics and applications, October 1983, p.p. 25–37.

[2] Pratt M.J. 1990 "Solid Modelling–Survey and Current Research Issues" in "Computer Graphics Techniques Theory and Practice" editors: David F. Roger, Rae A Earnshaw, p.p. 363–405, 1990 by Springer Verlag

[3] Sabin, Malcom 1990"Interrogation Techniques for Parametric Surfaces" in "Computer Graphics Techniques Theory and Practice" editors: David F. Roger, Rae A Earnshaw, p.p. 339–361, 1990 by Springer Verlag

[4] Teixeria, J.C. 1989"Interface with Geometric Modelling Systems Based on Constructive Geometry" Proceeding of the IFIP TC 5 Conference on CADCAM Technology Transfer: Applications of computers to Engineering design, manufacturing and management, 1989, Mexico, edited Lastra, G.L. et al, p.p. 39–50

[5] Encarnacao, J.L, Schonhut, J. 1989"High performance, visualisation and integration: the computer graphics headlines for 90's" Proceeding of the IFIP TC 5 Conference on CADCAM Technology Transfer: Applications of computers to Engineering design, manufacturing and management, 1989, Mexico, edited Lastra, G.L. et al, p.p. 31–38

[6] STEP,1993 "Product Data Representation and Exchange–Part 1: Overview and Fundamental Principles", *ISO CD 10303–1*, 1991

[7] Mantyla, Martti 1989 "Directions for Research in product Modeling." *Proc. Third Int. IFIP Conf. on Comp. Applic. in Prod. and Eng. CAPE* 89. Tokyo, Japan. 1989.

[8] Mok, H.H., 1992 "Feature recognition using Parasolid" M.Sc. thesis, Loughborough University of Technology.

[9] Krause, F.–L et al 1993"Product Modelling"Annals of the CIRP Vol.42/2/1993, pp. 695–706.

[10] Parasolid, v5.0, 1992 "Programming Reference Manual" *Electronic Data Systems Corporation*, 13736 Riverport Drive Maryland Heights, MO 63043

[11] Objectivity 1991 "Objectivity/DB C++ Programming Interface Manual" *Objectivity*, Inc.

[12] Booch, G. 1993 "Object Oriented Design With Applications" *The Benjamin/Cumings Publishing Company*

[13] IDEF, 1990 "User Manual" Meta Software 1990.

[14] The EXPRESS, 1993 "Language Reference Manual" Document ISO DIS 10303–11.