

**DEVELOPMENT OF A WEB-BASED
MANUFACTURING APPLICATION SYSTEM
FOR ROTATIONAL PARTS**

**A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY**

BY

ERHAN ÖZSÜER

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

THE DEPARTMENT OF MECHANICAL ENGINEERING

DECEMBER 2003

Approval of the Graduate School of Natural and Applied Sciences.

Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Prof. Dr. Kemal İDER
Head of the Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Prof. Dr. Ömer ANLAĞAN
Supervisor

Examining Committee Members

Prof. Dr. S. Engin KILIÇ (Chairman)

Prof. Dr. Ömer ANLAĞAN

Prof. Dr. Metin AKKÖK

Prof. Dr. Mustafa İ. GÖKLER

Assoc. Prof. Dr. Tayyar ŞEN

ABSTRACT

DEVELOPMENT OF A WEB-BASED MANUFACTURING APPLICATION SYSTEM FOR ROTATIONAL PARTS

Özsüer, Erhan

M. Sc., Department of Mechanical Engineering

Supervisor: Prof. Dr. Ömer Anlağan

December 2003, 114 pages

Developing process plans and part programs rapidly and correctly for CNC machine tools plays a vital role in manufacturing. This study is concerned with the development of a web-enabled virtual design and manufacturing application system for rotational parts. The object oriented methodology is used in the application development. Windows Distributed interNet Application (DNA) architecture which describes a framework of building software technologies in an integrated web and client-server model of computing, is employed in the system. The entire system was modeled with Unified Modeling Language (UML), which is an industry-standard visual modeling notation to express software development architectures. Feature-based design approach, being a practical way of linking the design with manufacturing, is implemented in the rotational part design.

Users have to be registered in order to use the system. With the supplied web site, users can easily register to system. After registration, the user obtains a password and a unique username. Upon the user authentication, the user session starts. A typical user session involves new or past project selection, material selection, part sketching, blank size definition and cutting parameters determination.

After all the steps are completed, process plan and part-program, which are required to manufacture the part on Denford Mirac CNC Turning Machine are generated and displayed on the web site and then saved to the database. The application enables the users to see past projects and to generate new process plan and part programs for different cutting parameters. The process plan, part program and the 2D wire frame drawing of the corresponding part are demonstrated on the web site for the chosen projects.

Keywords: Web-based Design and Manufacturing, Distributed Design and Manufacturing, Concurrent Engineering, Feature-Based Design, DNA, OOP

ÖZ

DÖNEL PARÇALAR İÇİN WEB TABANLI BİR İMALAT UYGULAMA SİSTEMİNİN GELİŞTİRİLMESİ

Özsüer, Erhan

Yüksek Lisans, Makina Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Ömer Anlağan

Aralık 2003, 114 sayfa

İmalatta, bilgisayar kontrollü sayısal tezgahlar için doğru ve hızlı bir şekilde işlem planı ve parça programı üretmek oldukça önemlidir. Bu çalışma, dönel parçalar için web tabanlı sanal bir tasarım ve imalat sisteminin geliştirilmesini amaçlamaktadır. Uygulamada nesne yönelimli yazılım yöntemi ile sunucu ve istemci taraflı ağ yazılımları geliştirmek için yapısal çözüm yolları sunan Windows Dağıtık İnternet Mimarisi (DNA) kullanılmıştır. Yazılım mimarisi, endüstri standardı olmuş görsel bir modelleme dili olan UML ile modellenmiştir. Dönel parçaların tasarımında, tasarım ve imalat işlemleri arasındaki geçişi hızlandırıp, kolaylaştıran unsur tabanlı parça tasarım yöntemi kullanılmıştır.

Sistem kullanımı için kullanıcılara kayıt zorunluluğu getirilmiştir. Sağlanan web sitesiyle kullanıcılar sisteme kayıt olup bir kullanıcı ismi ve parola sahibi olurlar. Kayıtlı kullanıcılar sahip oldukları kullanıcı ismi ve parolayla site oturumunu başlatırlar. Tipik bir kullanıcı oturumu yeni bir projeye başlamayı veya eski bir projeyi görüntülemeyi, malzeme seçimini, parça tasarımını, ham madde boyutu

seçimini ve kesme parametrelerinin belirlenmesini içerir. Tüm bu adımlar tamamlandıktan sonra Denford marka sayısal tornada işlemek için gerekli olan işlem planı ve parça programı üretilir. Üretilen işlem planı ve parça programı web sayfasında görüntülenir, kullanıcı ve parça ilişkili olarak da veri tabanına kayıt edilir. Uygulama, kullanıcıya daha önceden tasarladığı parçaların resmini, işlem planını ve parça kodunu görmesine ve bu parçalar için farklı kesme parametreleri tanımlayarak yeni işlem planı ve parça programı üretmesine de olanak sağlar.

Anahtar kelimeler: Web tabanlı tasarım ve imalat, Dağıtık tasarım ve imalat, Eş Zamanlı Mühendislik, Unsur Tabanlı Tasarım, DNA, OOP

To my lovely family

ACKNOWLEDGMENTS

I would like to thank my thesis supervisor Prof. Dr. Ömer Anlağan for his continuous support and guidance throughout my work.

I wish to express my gratitude to Prof. Dr. S. Engin Kılıç for his close attention and encouragement during my study.

It is impossible for me to forget the understanding and kindness of Prof. Dr. Sinan Kayalıgil while assisting him in Manufacturing Modeling and Simulation Laboratory in Industrial Engineering Department and doing my thesis work together.

I am also thankful to our CIMLAB crew: Yusuf Başıbüyük, Saleh M. Amaitik, Burak Sarı, Umut Durak, Başar Budak, Boran Alataş and Fatih Sarı.

Finally of course my family, without their never ending support this work would not be possible.

TABLE OF CONTENTS

ABSTRACT.....	iii
ÖZ	v
ACKNOWLEDGMENTS.....	viii
TABLE OF CONTENTS.....	ix
LIST OF FIGURES.....	xiii
LIST OF TABLES	xvi
LIST OF SYMBOLS	xvii
LIST OF ACRONYMS.....	xviii
CHAPTER	
1. INTRODUCTION.....	1
1.1 About CIM.....	2
1.2 Historical Developments in Advanced Manufacturing Technologies.....	3
1.3 Research Trends in CIM and Related Technologies.....	4
1.4 Application Architecture.....	5
1.4.1 Single-tier Applications.....	6
1.4.2 Two-tier Applications	6
1.4.3 Multi-tier Applications.....	7
1.5 Component Object Model + (COM+)	8
1.6 Microsoft SQL Server.....	8
1.7 Active Server Pages (ASPs).....	8
1.8 Scope & Outline.....	9
2. LITERATURE SURVEY.....	11
2.1 Concurrent Engineering.....	11
2.2 Virtual Manufacturing	12

2.3 CAE, CAD, CAM.....	12
2.4 Machining Process Planning.....	13
2.5 Process Planning Integration Framework	15
2.5.1 Characteristics of Process Planning Function	16
2.5.2 Specification of Process Planning Systems.....	16
2.6 Computer-Aided Process Planning	17
2.6.1 Variant Approach	18
2.6.2 Generative Approach.....	18
2.7 CAPP: Implementation Techniques.....	19
2.7.1 Decision Tables	20
2.7.2 Decision Trees.....	20
2.7.3 Expert Systems Techniques (AI).....	21
2.8 CAD/CAM Integration and CAPP Features	21
2.9 Feature-Based Design	22
2.9.1 Feature Recognition and Extraction Systems (FRES)	23
2.9.2 Feature-Based Modeling Systems (FBMS).....	24
2.9.3 Feature-Based Design Systems (FBDS).....	26
2.10 CNC Part Programming	27
2.11 Microsoft Distributed interNet Applications Architecture (DNA).....	29
2.12 Object Oriented Programming (OOP)	30
2.13 Unified Modeling Language (UML)	32
2.13.1 UML Use Cases and Use Case Diagrams	33
2.13.2 UML class diagrams and static modeling	34
2.13.3 Dynamic Modeling.....	34
2.13.4 State Diagram	34
2.13.5 Activity Diagram.....	35
2.13.6 Sequence Diagram.....	35
2.13.7 Collaboration Diagram.....	35
2.14 Overview of the Studies on CAD/CAM, CAPP	36
2.15 Overview of the Studies on Internet-Based Design and Manufacturing	41

3. PROPOSED SYSTEM STRUCTURE.....	44
3.1 Design Features Description and Product Data Representation	45
3.2 Geometric Part Design.....	45
3.3 Interpretation of Product Data File	48
3.4 Process Planning Procedure.....	48
3.5 Machining Calculations	49
3.6 Part Programming	52
3.7 Machine Tool Offsets	53
3.8 Basic G and M Codes in Part Programming.....	54
3.8.1 G Codes (Preparatory Functions).....	54
3.8.2 M Codes (Miscellaneous Functions).....	56
3.9 Cutting Tools	56
4. APPLICATION DEVELOPMENT	59
4.1 Presentation Tier	59
4.2 Business Logic Tier	60
4.3 Data Tier	60
4.4 Part Design Control Development.....	64
4.5 Reading of Product Data File.....	70
4.6 Process Planning	71
4.7 Part Programming	71
5. TEST RUNS	72
5.1 Sample Run 1	72
5.2 Sample Run 2 (manufactured)	80
5.3 Sample Run 3	86
6. CONCLUSION AND FUTURE WORK.....	95
REFERENCES	98
APPENDICES	
A. USER GUIDE.....	104
B. UML DIAGRAMS	107
B.1 Use Case Diagram.....	107
B.2 Activity Diagram.....	108

C. G CODES.....	109
C.1 Sample Run 1	109
C.2 Sample Run 2	110
C.3 Sample Run 3	111
D. DENFORD MIRAC CNC TURNING MACHINE SPECIFICATIONS...	113

LIST OF FIGURES

FIGURE

1.1 Present market condition [1]	1
1.2 Development of manufacturing technologies based on information techniques [6]	4
1.3 User interface, business rules, and database reside separately [10]	7
2.1 Conventional process planning in manufacturing systems [21]	15
2.2 Variant approach to CAPP [21]	18
2.3 Generative approach to CAPP [21]	19
2.4 Decision table and tree [21]	20
2.5 Feature recognition and extraction system (FRES) [21]	24
2.6 Feature-based modeling system (FBMS) [21]	25
2.7 Feature-based design system (FBDS) [21]	27
2.8 DNA architecture	30
2.9 Elements of Use Case diagram [29]	33
2.10 Expert system basic aspects [41]	41
2.11 Architecture of the CyberCut Service [43]	42
3.1 A sample part drawing, comprised of five external features and its data file.	47
3.2 A sample part drawing, comprised of six external and two internal features and its data file.	47
3.3 Steps in pocket identification procedure	51
3.4 Nodal points on a sample part drawing	51
3.5 Working envelope of the CNC lathe	52

3.6 Diameter and radius programming.....	53
3.7 Movement of G00	54
3.8 Movement of G01	54
3.9 Movement of G02	55
3.10 Movement of G03	55
3.11 Design features for rotational parts	58
4.1 DNA Architecture	59
4.2 Web site diagram of the application.....	60
4.3 Database diagram of WebTURN	62
4.4 DNA Model of WebTURN.....	63
4.5 UML class diagram of the business logic	63
4.6 ActiveX component and data entrance window.....	65
4.7 Sketching and taper feature data entrance windows	67
4.8 Conditions of the convex & concave features.....	68
4.9 Data entrance window for convex and concave features	69
4.10 Data entrance window for drill feature	70
5.1 Run-1: Registration page.....	73
5.2 Run-1: Log in page.....	73
5.3 Run-1: New or past project selection page.....	74
5.4 Run-1: Part name entry page.....	74
5.5 Run-1: Material selection page	75
5.6 Run-1: Engineering drawing of the part (units in mm).....	76
5.7 Run-1: Sketched part in the design stage	76
5.8 Run-1: Billet definition and cutting parameters edit page	78
5.9 Run-1: Process plan page	79
5.10 Run-1: Part program page	80
5.11 Run-2: Part name entry page.....	81
5.12 Run-2: Material selection page	81
5.13 Run-2: Engineering drawing of the designed part (units in mm).....	82
5.14 Run-2: Sketched part in design stage	82
5.15 Run-2: Billet definition and cutting parameters edit page	84
5.16 Run-2: Process plan page	85

5.17 Run-2: Part program page	86
5.18 Run-3: Log in page.....	87
5.19 Run-3: New or past project selection	87
5.20 Run-3: Material selection	88
5.21 Run-3: Part name entry page.....	88
5.22 Run-3: Engineering drawing of the part (units in mm).....	89
5.23 Run-3: Sketched part in design stage	89
5.24 Run-3: Billet definition and cutting parameters edit page	91
5.25 Run-3: Process plan page	92
5.26 Run-3: Part program page	93
5.27 Run-3: Past project view	94
A.1 WebTURN home page	104
A.2 MS Windows 2000 components services window-1	105
A.3 MS Windows 2000 components services window-2	106
B.1 Use case diagram.....	107
B.2 Activity diagram.....	108

LIST OF TABLES

TABLE

3.1 Taxonomy of rotational features	46
3.2 Equations for cutting operations	50
3.3 Cutting tools and tool index numbers	56
4.1 Materials and chosen cutting parameters	61
5.1 Run-1: Part data file	77
5.2 Run-2: Part data file	83
5.3 Run-3: Part data file	90

LIST OF SYMBOLS

a_p	depth of cut (mm)
d_m	diameter of machined surface (mm)
f	feed rate (mm/rev)
l_w	length of surface or hole to be machined (mm)
n_t	rotational speed of tool (rpm)
n_w	rotational speed of workpiece (rpm)
η_m	efficiency of the machine tool
P_s	specific cutting energy (GJ/m ³)
Z_w	metal-removal rate (m ³ /s)

LIST OF ACRONYMS

AI	Artificial Intelligence
AMT	Advanced Manufacturing Technology
ASCII	American Standard Code for Information Exchange
ASP	Active Server Pages
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CAPP	Computer Aided Process Planning
CE	Concurrent Engineering
CGI	Common Gateway Interface
CIM	Computer Integrated Manufacturing
CNC	Computer Numerical Control
COM	Component Object Model
DLL	Dynamically Linked Library
DNA	Distributed interNet Architecture
FBDS	Feature-Based Design System
FBMS	Feature-Based Modeling System
FBS	Feature Based System
FRES	Feature Recognition and Extraction System
GT	Group Technology
HTML	Hypertext Markup Language
LM	Lean Manufacturing
NC	Numerical Control

OOP	Object Oriented Programming
PERL	Practical Extraction Report Language
RDBMS	Relational Database Management System
SQL	Structured Query Language
STEP	Standard for Exchange of Product model data
UML	Unified Modeling Language
VM	Virtual Manufacturing
WWW	World Wide Web

CHAPTER 1

INTRODUCTION

Manufacturing enterprises play an important role in improving the economic environment of a country. Today, the capability of producing high quality products with shorter delivery times and the ability to produce according to the diverse customer requirements have become the required characteristics for manufacturing industries. Hence, application of intelligent manufacturing systems and computer integrated manufacturing (CIM) have become the necessity to overcome the above issues while increasing the employment level and revenue of a country in ever changing global market. With the developments taking place in CIM and its related technologies, the application of CIM in manufacturing enterprises has become a reality [1]. Figure 1.1 shows the present market condition.



Figure 1.1 Present market condition [1]

CIM is a management and manufacturing strategy. CIM applications are becoming common place in manufacturing industries. Concurrent engineering, lean manufacturing, responsive manufacturing, agile manufacturing, virtual manufacturing are some of the new terms evolved in the last decade to reflect the dynamic nature of improvements in manufacturing applications. However, the concept of CIM is far broader than these new terms and it is still able to embrace all the new features of similar strategies. CIM stands as the innovative application for yesterday's proposals and the newer application required today and for the future [1].

1.1 About CIM

The concept of CIM was initially invented by Harrington in 1973 in the book "Computer Integrated Manufacturing" [2]. However, until the early 1980s, CIM did not become a commonly known acronym, as it exists today. CIM has a fundamental strategy of integrating manufacturing facilities and systems in an enterprise through computers and its peripherals. CIM has various meanings depending on the application of it. CIM is a process of using computers and communication networks to transform islands of enabling technologies into a highly interconnected manufacturing system. CIM involves integration of advanced technologies in various functional units of an enterprise in an effective manner to achieve the corporate objective of the manufacturing enterprise. Effective integration requires an in-depth understanding of all activities in all functional units of an enterprise. To be successful in the application of CIM, first the integration of advanced manufacturing technologies (AMTs) should be achieved, the computers act only as subordinates to the technologies. However, without computers the integration cannot be effective and efficient since computers help organize, retrieve and store information in an orderly manner with high accuracy and speed [1].

1.2 Historical Developments in Advanced Manufacturing Technologies

Mechanization facilitated mass production to meet the consumers' demands for improved products. To achieve mass production transfer lines and fixed automation were created. This resulted in the development of programmable automation [3]. The major objective of the automation was to accelerate the production process throughout the plant. Automation has provided the ability to respond to the customers' demand while ensuring high quality products. Numerical control was developed as an innovative approach to programmable automation in 1952 and gave a tremendous boost to the concept of mechanization and automation [4]. With the developments in commercially available computer technology, the application of computers in manufacturing started to emerge producing a variety of new technologies, all of which are collectively named as AMTs. AMTs are generally defined as systems providing flexibility as well as data driven computer integration for a manufacturing organization [5]. AMT illustrates a technological application where the manufacturing technology utilized is intelligent enough to process the activities with less human intervention. In today's industrialized world, the enormous growth in manufacturing automation has brought diverse AMTs with number of features. These AMTs could consist of semi to fully automated systems or equipments.

The major element in an integration concept is having a common or interconnected database, by which the data can be shared automatically among various units and user groups. By introducing an integrated manufacturing concept, simplified workflow and reduction in the work-in-progress will be the standard and beneficial features. The integration of the technologies brings the following benefits [1]:

1. Creation of a truly interactive system, which enables manufacturing functions to communicate easily with other relevant functional units.
2. Accurate data transferability among the manufacturing plant or subcontracting facilities at in-plant or diverse locations.

3. Faster responses to data-changes for manufacturing flexibility.
4. Increased flexibility towards introduction of new products.
5. Improved accuracy and quality in the manufacturing process.
6. Improved quality of the products.
7. Control of data-flow among various units and maintenance of user-library files for system-wide data.
8. Reduction of lead times which generates a competitive advantage.
9. Streamlined manufacturing flow from order to delivery.
10. Easier training and re-training facilities.

1.3 Research Trends in CIM and Related Technologies

To meet the demands of the market, research trends focusing on the application of CIM in manufacturing industries has begun to emerge. It is realized that CIM requires effective integration of a number of available AMTs. Advanced manufacturing technologies rely heavily on various information techniques to achieve higher productivity, higher productivity and lower production costs. In the past five decades manufacturing industry has experienced some notable changes in the information technology, manufacturing automation and management information system. Figure 1.2 shows this historical development.

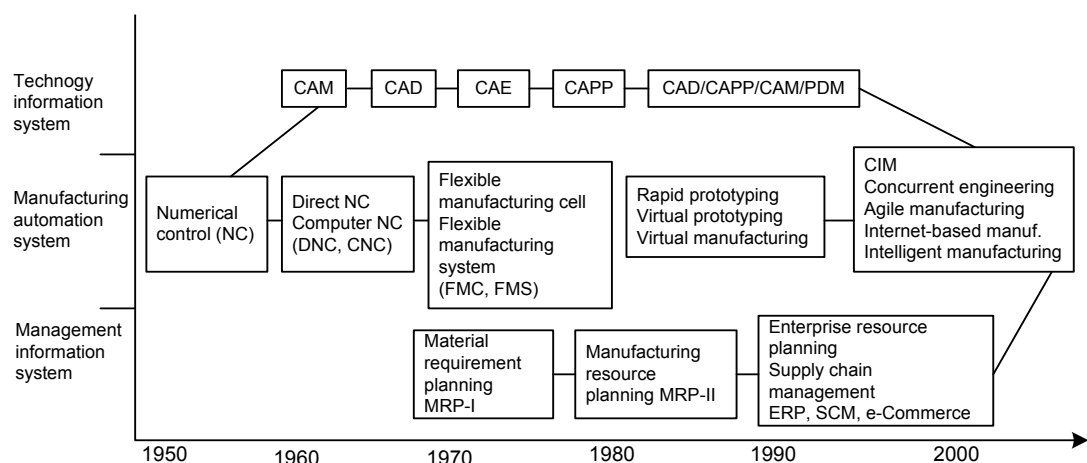


Figure 1.2 Development of manufacturing technologies based on information techniques [6]

The concept of concurrent engineering (CE) has existed in manufacturing from early 1960s in different forms requesting the use of multidisciplinary teams to

accelerate product introduction. However, in 1987, the concept was given the name “concurrent engineering” with an appropriate definition by the United States Defense Advanced Research Projects Agency. CE involves, a systematic and simultaneous approach to the integrated design of products and their related processes including marketing, manufacturing, sales and purchasing [7]. Further it involves formation of multidisciplinary teams for the rapid product development and introduction of the product into the market. CE could be considered as a management strategy rather than the manufacturing strategy.

In the late 1950s, Japanese automobile manufacturers realized that the mass production did not fit into the production and management strategy required to satisfy the product differentiation, which was evolving as a fundamental market feature [8]. This resulted in the formation of the concept, lean manufacturing (LM). LM involves addressing the product strategy, product development, supply chain, manufacturing and product distributions for the production of diverse products in small batches. It may further consist of out-sourcing and effective integration between the suppliers and subsidiaries. From the early 1980s, the North American automobile manufacturers have begun to use LM concepts.

The concept of agile manufacturing is similar to the concept of LM by emphasizing small batch sizes. However, agility requires: reduction in product development time; allowance for considerable customization of product features, and incorporating highly adaptive, flexible and efficient manufacturing practices in the product development and manufacturing cycle [9]. It further involves effective integration of technology, business activities, enterprise and people.

1.4 Application Architecture

The system architecture is an important feature in application design because it defines how elements in the application interact and what functionality each element provides. The three types of system (or application) architecture are single-tier (or monolithic), two-tier, and multi-tier. Multi-tier applications can be

implemented using multiple computers across a network. This type of application is referred to as a distributed application, or *n-tier application*. An n-tier application represents a special instance of a three-tier application in which one or more of the tiers are separated into additional tiers, providing better scalability [10].

1.4.1 Single-tier Applications

In a single-tier application, only one layer supports the presentation, application logic, and data services. Only one application or application element processes all three of these services. The data itself can be physically stored in any location, such as on a server. However, the functionality for accessing the data is part of the application [10].

1.4.2 Two-tier Applications

Two-tier, or standard client/server applications, group presentation and application logic components on the client machine and access a shared data source using a network connection. In a two-tier application, the user interface and business rules exist on a single layer that runs on the client computer. Separate applications, such as SQL Server or Oracle database servers, provide the data services. Classical two-tier architectures have brought efficiencies to businesses, but there are also a number of limitations:

Difficult to scale: Application scaling is poor because of the limited number of database connections available to clients. Connection requests beyond this limit are simply rejected.

Difficult to maintain: It is hard to maintain client-side application logic because it has to be deployed to every client. Any change in the logic must be redistributed to all clients.

Poor network performance: A network runs inefficiently because of the amount of raw data that is transferred across it. Much of the database processing is not localized [10].

1.4.3 Multi-tier Applications

Tiers are a logical concept. The three tiers are generally described as user (first), business (second or middle), and data (third); however, there can be more than three tiers in a multi-tier application. Because of this fact, multi-tier applications are sometimes referred to as n-tier applications where n is any number greater than or equal to three. A service is a unit of application logic that implements operations, functions, or transformations that are applied to objects. For example, a business service can be implemented as a COM component that ensures a purchase does not exceed the buyer's credit limit. In multi-tier architectures, presentation, application logic, and data elements are conceptually separated. These tiers don't necessarily correspond to physical locations on the network. For example, all three tiers may exist on only two machines or they may be deployed on five.

Presentation components manage user interaction and request application services by calling middle-tier components. Application components perform business logic and make requests to databases.

With multi-tier applications, the client provides only one layer: the user interface. The business rules are performed on a system between the user interface and the data storage system. This allows the presentation services, user interface, business rules, and database to reside separately as shown in Figure 1.3 [10].

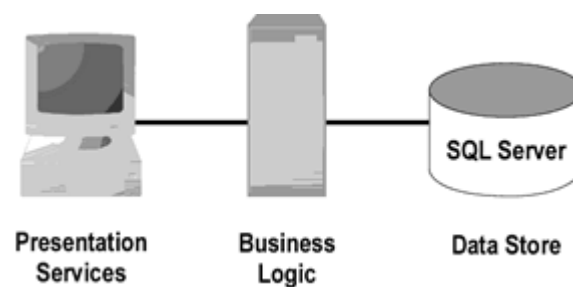


Figure 1.3 User interface, business rules, and database reside separately [10]

1.5 Component Object Model + (COM+)

COM+ is an extension of Component Object Model (COM), Microsoft's strategic building block approach for developing application programs. COM+ is both an object-oriented programming architecture and a set of operating system services. It adds to COM a new set of system services for application components while they are running, such as notifying them of significant events or ensuring they are authorized to run. COM+ is intended to provide a model that makes it relatively easy to create business applications that work well with the Microsoft Transaction Server (MTS) on a Windows NT or a subsequent system [11].

1.6 Microsoft SQL Server

Microsoft SQL Server is a relational database management system (RDBMS). A relational database provides a means of organizing data by storing it in database tables. All data can be grouped into tables, and relationships can be defined between tables as well—thus the name, relational database. Users access the data on the server through an application. Administrators access the server directly to perform configuration, administrative, and database maintenance tasks. SQL Server is a scalable database, which means that it can store substantial amounts of data and can support many users accessing the data simultaneously.

Its primary purpose is to serve as back-end database for a client program, such as a web browser, an accounting program or a human resource application, anything that makes use of the data. In the most common usage scenario, a client program connects to SQL Server and requests some information, and then the SQL Server processes the request and returns results [12].

1.7 Active Server Pages (ASPs)

An Active Server Page is a collection of server-side scripts and HTML tags in files that reside on the server. The scripts can communicate with components

running on local server or any other servers. ASP files requested by a browser are loaded and processed by the server and as a result, page is returned to client as a standard HTML. The followings are possible by ASP programming [13]:

- Dynamically edit, change or add any content of a Web page
- Respond to user queries or data submitted from HTML forms
- Access any data or databases and return the results to a browser
- Customize a Web page to make it more useful for individual users
- The advantages of using ASP instead of CGI (Common Gateway Interface) and PERL (Practical Extraction Report Language), are those of simplicity and speed
- Provides security since ASP code can not be viewed from the browser
- Since ASP files are returned as plain HTML, they can be viewed in any browser
- Clever ASP programming can minimize the network traffic.

1.8 Scope & Outline

The motivation of this study is to develop a distributed, feature-based design system for rotational parts and to generate process plan and part program for the designed part, which are specific to the Denford Mirac CNC Turning Machine by implementing the essentials of Microsoft DNA.

Both the ActiveX control and dynamically linked libraries (DLLs) for the server business processing are coded in Visual Basic 6. SQL Server 2000 is the chosen relational database management system. HTML and ASP scripting are used in website coding. Macromedia Dreamviewer MX is utilized as the website editor.

The thesis outline is: second chapter covers the literature survey on up-to-date manufacturing concepts and their implementations. Chapter 3 documents the system model, and building parts of the thesis namely feature-based design, process planning and part programming. Chapter 4 describes the development procedure of

the WebTURN. Chapter 5 is about the test runs of the application. Two full samples are shown and described in this chapter. Chapter 6 concludes the work and provides recommendations for future works.

CHAPTER 2

LITERATURE SURVEY

2.1 Concurrent Engineering

Concurrent Engineering (CE) is a product development approach that has been previously used in electronic and manufacturing system design. The objective of CE is to reduce the system/product development cycle through a better integration of activities and processes. A more formal definition has been provided by the Institute of Defense Analyses (IDA): “Concurrent Engineering is the systematic approach to the integrated, concurrent design of products and related processes, including manufacturing and support. This approach is intended to cause the developers to consider all elements of the product life cycle from conception through disposal, including quality, cost, schedule, and user requirements.”

The basic principle of CE is the integration of methodologies, processors, human beings, tools, and methods to support product development. CE is multi-disciplinary in that it includes aspects from object-oriented programming, constraint programming, visual programming, knowledge-based systems, hypermedia, database management systems, and CAD/CAM.

Concurrent Engineering involves the interaction of diverse group of individuals who may be scattered over a wide geographic range. To enable effective and complete communication among them, there are certain technological concepts that must also become organized into concurrent layers. Distributed information sharing and collaborative/cooperative work are important techniques to maintain or exceed

the current level of software development productivity. Concurrent Engineering takes advantage of shared information and allows simultaneous focus on different phases of the software development life cycle. Many existing World Wide Web (WWW) capabilities could support a wide area CE environment [14].

2.2 Virtual Manufacturing

Virtual manufacturing (VM) and virtual engineering have been identified as one of the enabling technologies of agile manufacturing and its related activities [15, 16]. One definition of virtual manufacturing is “to carry out manufacturing activities with a simulation model of actual setup, which may or may not exist. It holds all the information relating to the process, the process control and management and product specific data. It is also possible to have part of the manufacturing plant be real and the other part virtual” [17]. From such a definition, it can be inferred that integration of simulation models with process control and management data and possibly with real system itself is an essential ingredient within such an approach [18].

2.3 CAE, CAD, CAM

Computer Aided Engineering (CAE), in the broadest sense of the meaning, covers any application of computers to the solution of engineering problems and this whole field of computer applications is one that expands continually, slowly moving towards the day when a total concept in CAE will be possible.

Computer Aided Design (CAD) refers to any application of computers to the solution of design problems. However, the generally accepted and somewhat restricted meaning is the application of computers to engineering design problems wherein an extensive use is in field of computer graphics. The aim of CAD is to apply computers to both the modeling and communication of designs. There have been two different approaches, which are often used together:

1. At a basic level, to use computers to automate or assist in such tasks as the production of drawings or diagrams and the generation of list of parts in a design.
2. At a more advanced level, to provide new techniques that give the designer enhanced facilities to assist in the design process.

Computer Aided Manufacturing (CAM) refers to any application of computers to the solution of manufacturing problems. The impact of computers in manufacturing industry is more evident in the form of NC and CNC machine tools. For particular applications, it is possible to further increase productivity by grouping a variety of these NC machines together with an automated materials handling system to form a manufacturing cell controlled by a computer. This is referred to as flexible manufacturing system (FMS).

In the typical CAD/CAM system, the user interacts with the computer using a graphics terminal, designing and manufacturing a part with information stored in the computer database [19]. The user may:

- Construct a geometric model
- Analyze the structure
- Perform kinematic studies
- Produce engineering drawings
- Create instructions for machine tools
- Produce process plans for fabrication.

2.4 Machining Process Planning

Machining process planning generally defined as the system responsible for converting the design specification of a product into a set of manufacturing instructions within a machining process such as metal cutting. Process planning is also a function within a manufacturing facility that establishes the machining process, identifies the machines capable of performing these processes and sets the machining parameters in order to convert raw material into a desired form, as

specified in an engineering drawing. Many machining processes are available e.g. shaping, drilling, boring, milling, turning and grinding. The resulting set of instructions for machining process planning are the followings:

1. Identification of design specifications,
2. Determination of design specifications,
3. Selection of machine tools to perform the required machining operations,
4. Design of jigs and fixtures,
5. Determination of setup requirements,
6. Choice of cutting tools,
7. Determination of operation sequences or routes,
8. Calculation of cutting parameters,
9. Generation of CNC programs,
10. Simulation of CNC programs.

A typical machining process planner is a well-trained and experienced machinist or production engineer who is not only familiar with the production process, but also the capabilities of various facilities on the factory floor. He or she has a large amount of accumulated knowledge for process planning by semi logical methods, such as recognizing a familiar pattern and recalling the appropriate machining processes.

Traditional process planning relies heavily on human decisions and manual calculations. Depending on the experience of the planner, manual planning may be highly labor intensive and susceptible to error. It is a time-consuming task and plans generated by planners are not always consistent. There is no guarantee of efficiency in the planning of machining sequences. The initial plan may be revised several times in an attempt to achieve feasibility. In general, machining process planning has the following characteristics [20]:

1. It is a very complicated task.
2. It requires good deal of professional expertise and knowledge.

3. It is time consuming.
4. It is not consistent.
5. It is very depended on resources.
6. It is difficult to use mathematical modeling.
7. It is difficult to obtain the optimum plan.

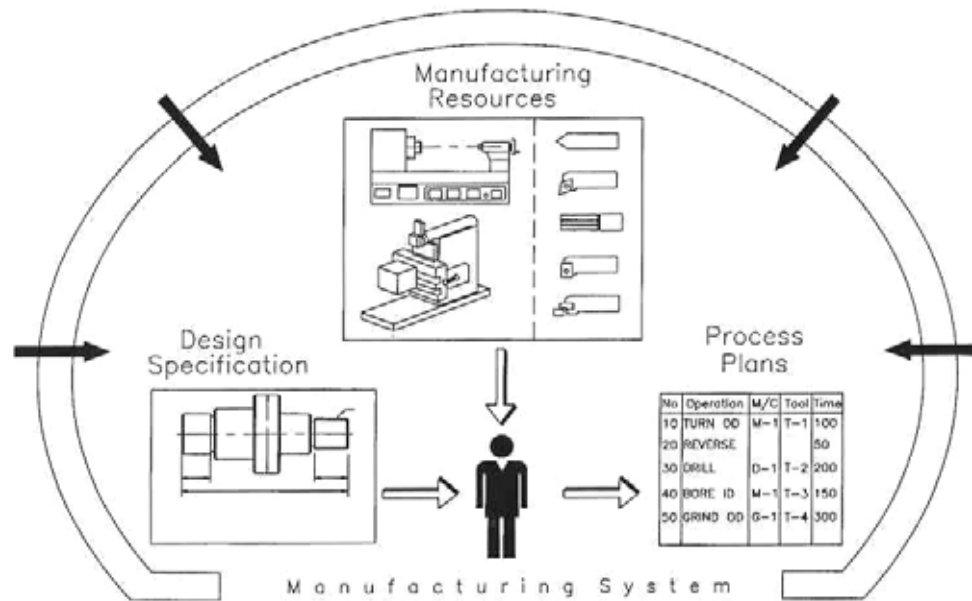


Figure 2.1 Conventional process planning in manufacturing systems [21]

2.5 Process Planning Integration Framework

A framework is a software architecture that defines an open environment for the development and integration of application software components or systems in a given context. A process planning integration framework provides an environment for process planning functions to be readily integrated with each other and product life cycle functions, such as product design or product scheduling [22].

2.5.1 Characteristics of Process Planning Function

Process planning is a complicated task, which can be characterized as follows [22]:

- Process planning task varies from company to company and from product to product. Process planning task varies for different domains. Process planning function can be classified with the following specification parameters: product domain, process domain, activity domain, planning mode domain, input mode and output resolution.
- The process planning function can be decomposed into a set of tasks that are relatively independent of each other. The flow of data between the tasks results in the task interacting.
- The process planning function evolves as new processes and resources are available. Process planning task is dynamic in nature because the planning strategies and data involved change as new materials, processes and equipment are introduced.

2.5.2 Specification of Process Planning Systems

In general, process planning systems can be specified by six domains:

1. *Product domain*

In manufacturing industries, there are a wide variety of product types, geometries, materials, etc. A process planning system often only deals with a specific type of product, geometry or material. For instance, there are individual process planning systems for rotational and prismatic parts.

2. *Process domain*

A process planning system is usually associated with a set of processes and technologies. For instance, there are individual process planning systems for machining, assembly and sheet metal processing.

3. *Activity domain*

The activity domain characterizes the activities, such as degree of automation, size of production facility involved, etc.

4. *Planning method domain*

Several different planning methods are used in process planning, such as variant, semi-generative, generative and artificial intelligence.

5. *Input mode*

Depending on the applications and level of automation, several input modes exist in CAPP. They include manual input, GT code, special part description language, CAD generated data, and standard neutral data file (such as a STEP Part file).

6. *Output resolution*

Output resolution refers to the level of detail in the generated process plan. The generated process plan can range from a rough process routing sheet to a very detailed process plan with operation/machining step descriptions and parameters.

For a process planning system, the system requirements can be defined by these domain specifications [22].

2.6 Computer-Aided Process Planning

Process planning establishes the methods and means of converting the raw material to a finished part in a manufacturing facility. Computer-aided process planning (CAPP) is the application of computers to assist the process planners in the planning function. CAPP is considered a crucial link between CAD and Computer-Aided Manufacturing (CAM) (e.g., production/scheduling systems). Over 30 years of research into CAPP has resulted in a wealth of knowledge. As a result, many experimental and commercial CAPP systems have been developed. There are two basic approaches to CAPP: variant and generative [21].

2.6.1 Variant Approach

The variant approach, which is also called retrieval approach, uses a group technology (GT) code to select a generic process plan from the existing master process plans developed for each part family and then edits to suit the requirement of the part. Variant approach is usually implemented with group technology coding system. Here, the parts are segmented into groups based on similarity and each group has a master plan. Compared to manual planning, it increases the information management capabilities moderately. But the main disadvantage of the variant process is that it still requires an expert planner to edit and modify the plans for specific components. The computer is just a tool to assist in manual process planning. Some examples using this approach are CAPP (Link, 1976), MIPLAN (TNO, 1981), AUTOCAP (El-Midany and Davies, 1981) and TOJICAP (Zhang and Gao, 1984) [21].

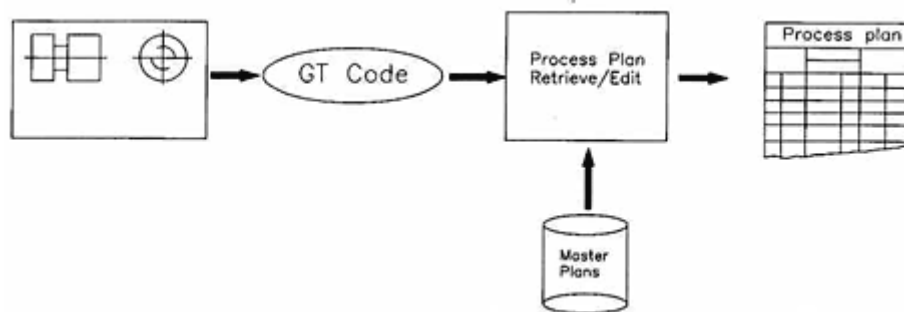


Figure 2.2 Variant approach to CAPP [21]

2.6.2 Generative Approach

In a generative approach, a process plan for each component is created from scratch without human intervention. These systems are designed to automatically synthesize process information to develop a process plan for a part. These systems contain the logic to use manufacturing database and suitable part description schemes to generate a process plan for a particular part. Most of the contemporary CAPP systems being developed are generative in nature. Generative approach eliminates disadvantages of variant approach and bridges the gap between the CAD and CAM [21].

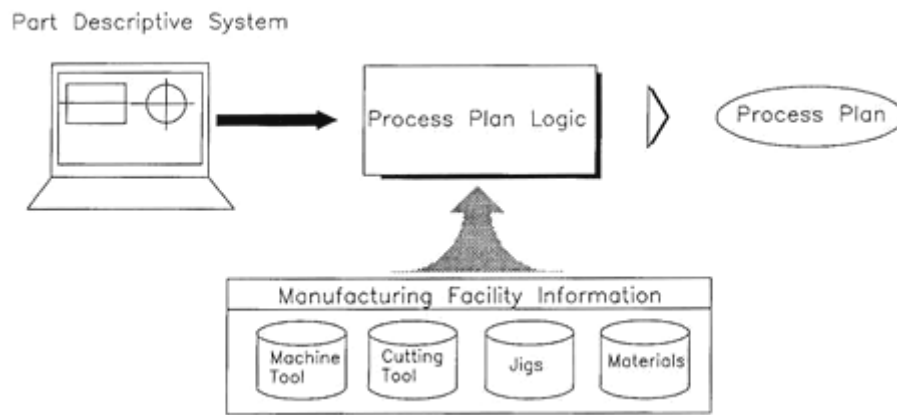


Figure 2.3 Generative approach to CAPP [21]

2.7 CAPP: Implementation Techniques

Logical decision is a traditional implementation technique used in CAPP. The simplest approach is to code the process capability in a computer program. A tree-structured classification can be used in the system and each process can be coded as a branch of a decision tree (Chang and Wysk, 1984).

The decision logic should be present in a format that is easy to visualize and check for completeness, contradictions, and redundancy. Generally, each manufacturing process is defined as a separate entity based on the capabilities to generate or modify geometric features or properties. Only values of decision variables change over time or between companies. The techniques for structuring the decision logic are numerous and varied.

The objective of decision logic in a CAPP system is to match the process capabilities with design specifications in an optimal way. Generally, most common decision logic can be classified as one of the three methods: decision tables, decision trees and AI [21].

2.7.1 Decision Tables

A decision table is partitioned into conditions and actions and is represented in a tabular form. It is program-structuring tool, which provides readable documentation as an automatic by-product. Also, decision tables can be used with preprocessors to eliminate some program coding, and to provide automatic checks for completeness, contradiction, and redundancy [21].

2.7.2 Decision Trees

A decision tree is a graph with a single root and branches emanating from the root. Decision trees are easier to customize, update, maintain, visualize and develop than decision tables. Decision trees can be represented as computer codes or data. The three as computer code is converted to a flow chart; the starting node is the root and every branch represents a decision statement, which is either false or true. Compared to decision tables, decision trees can be more easily updated and maintained. The flexibility in expanding and contrasting the tree, if necessary, is another advantage [21].

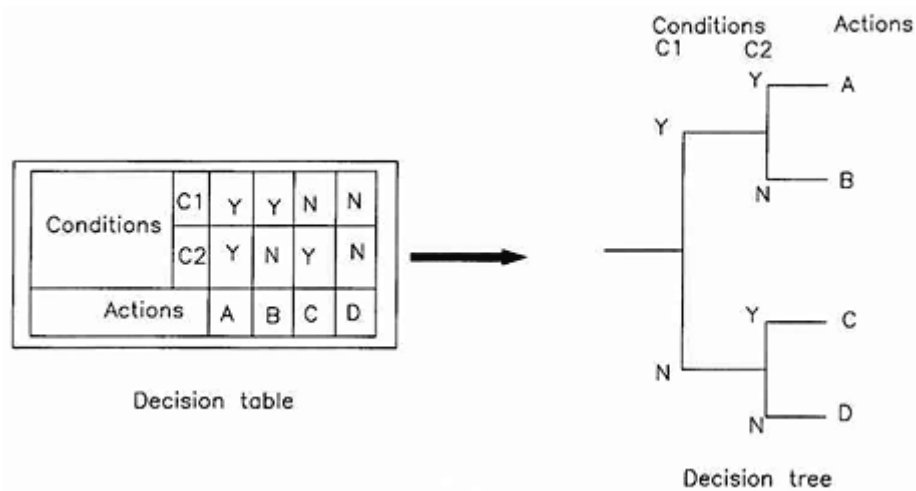


Figure 2.4 Decision table and tree [21]

2.7.3 Expert Systems Techniques (AI)

The solution of the process planning task depends on the empirical knowledge relevant to the organization, depending on the existing facilities. In general, problems in a production system can be represented by an initial state, a goal state, a set of operators, and a control structure. One of the frequently used methods to solve problems in AI is the theorem proving technique. In the part manufacturing problem, the initial state is the raw material or workpiece from which the part is to be produced and the goal state is the finished part. The set of operators comprises the available machine tools, cutters, etc. Processing of the part involves proceeding from the raw material to the finished part. Application of the operators (machine tools and cutters) moves the problem from one state to another. The various stages represent the workpiece in work-in-progress condition:

$$S_{\text{initial}} \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow \dots \rightarrow S_{\text{final}}$$

These stages are non-reversible as once the material is removed, it cannot be added back. However, in process planning, because the actual material removal has not been done, the stages can be reversed, leading to the development of process plans. This strategy is used in the development of expert CAPP system [21].

2.8 CAD/CAM Integration and CAPP Features

A frequently overlooked step in the integration of CAD/CAM is the process planning that must occur. CAD systems generate graphically oriented data and may go so far as graphically identifying metal, etc. to be removed during processing. In order to produce such things as NC instructions for CAM equipment, basic decisions regarding equipment to be used, tooling and operation sequence need to be made. This is the function of CAPP. Without some element of CAPP, there would not be such a thing as CAD/CAM integration. Thus CAD/CAM systems that generate tool paths and NC programs include limited CAPP capabilities or imply a certain approach to processing. CAD systems also provide graphically oriented data to CAPP systems to use to produce assembly drawings, etc. Further, this graphically oriented data can then be provided to manufacturing in the form of hardcopy

drawings or work instruction displays. This type of system uses work instruction displays at factory workstations to display process plans graphically and guide employees through assembly step by step. The assembly is shown on the screen and as an employee steps through the assembly process with a footswitch, the components to be inserted or assembled are shown on the CRT graphically along with text instructions and warnings for each step. If NC machining processes are involved, CAPP software exists which will select tools, feeds, and speeds, and prepares NC programs [23].

2.9 Feature-Based Design

Feature-based design provides the advantage of using high-level entity features, which are closely related to their manufacturing and assembly operations, as elements of a part. A feature, which represents a portion of a part, usually has a particular meaning or function for some design application.

From a designer's point of view, feature-based design offers possibilities for supporting the design process better than current CAD systems do. The implementation of the feature-based design method includes feature recognition, part representation, and design with features [24]. Features not only represent physical shapes, but can also describe other product information, such as material properties, technological parameters, and required manufacturing precision.

Feature recognition is the interpretation of a geometric model of an object to identify the relevant features and associated information. For a complete integration of CAD/CAM, feature recognition must be automated. In order to develop the feature-based model of a part, the features and their relationships can be recognized and extracted from an existing engineering drawing or from a CAD data file. These features and relationships carry information of great importance to the design activities of a product.

With feature-based design used as a key component of product representation in the whole product life cycle, all product development activities such as CAD, process planning and assembly planning, will be consistent and compatible with each other. Especially, for rotational parts, the geometric features have specific relationships that correspond to the manufacturing and assembly operations.

Feature-based systems (FBS) can be developed in three basic ways, each method have many variations. These can be listed as [21]:

- Feature recognition and extraction systems (FRES)
- Feature-based modeling systems (FBMS)
- Feature-based design systems (FBDS)

2.9.1 Feature Recognition and Extraction Systems (FRES)

In this approach, a feature recognition program examines the CAD database and makes the deductions about the types of features present. The additional responsibility of interpreting the part representation is shared either by a planning module or by a dedicated postprocessor. Hence, this can be treated as a post-design approach. The method amounts to make explicit what is implicit in a CAD database. Feature recognition systems work on a common principle of comparing predefined patterns with portions of the geometric model to retrieve the desired feature. Feature recognition can be based on volumes or surfaces, depending upon the definition adopted. Once the features are recognized, application oriented information, which was not included in the earlier stages of part representation, can be added to the features for completeness of the model, if necessary. However, the algorithms for recognizing even simple features are fairly complex and are generally modeler-specific. There are very few systems, with exceptions like CADEXCAP (Kalta and Davies, 1992), which can extract the technological (non-geometric) attributes.

Because of the complexities associated with FRES, it has become an attractive proposition to make use of features at the time of geometric modeling, which resulted in FBMS and FBDS. Moreover, proponents of other approaches argue that when a designer starts the design process, most of the features are already known; thus, in the process of converting them to a CAD model, the information is lost and has to be recreated by the feature recognizer (Joshi, 1990) [21].

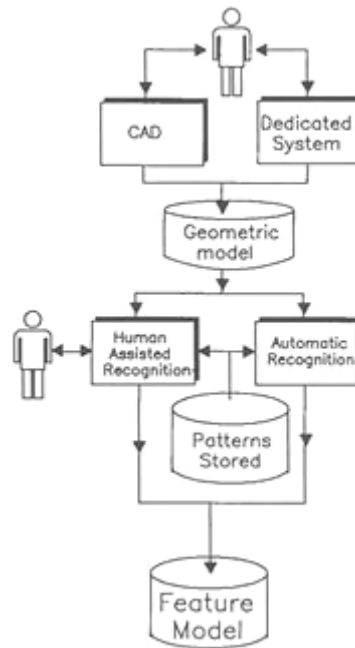


Figure 2.5 Feature recognition and extraction system (FRES) [21]

2.9.2 Feature-Based Modeling Systems (FBMS)

FBMS facilitates the interactive conversion of existing designs (usually in the form of drawing) to customize feature-based part models. This is also a post-design modeling method. This method helps in realizing a feature-based model in one step, as it is not necessary to create the geometrical model first. Nevertheless, a geometrical model can be created from the feature model for other purposes. This type of system plays a significant role in situations in which the part design is available only as a blueprint. In such cases, these can provide automated interfaces to CAPP by modeling the part in terms of features, without affecting the design practices followed in the industries. FBMS can be developed in two ways as shown in Figure 2.2.

1. *Using a dedicated system:* Some FBMS embrace system-specific and customized modeling environments. These systems are streamlined to the needs of the CAPP systems and may prove cost-effective, as they will not involve the use of any CAD systems. But the development of such systems involves a sizable amount of programming because typical geometry handling facilities are to be built completely from scratch.
2. *Using a front-end feature generator over a CAD system:* Modern CAD systems usually support an external high level language (like C) or an internal programming language (e.g. AutoLISP in AutoCAD, GRIP in UNIGRAPHICS) to enable the users to (a) access to internal CAD model and (b) customize the CAD system for different applications. These capabilities are used in developing FBMS by employing a front-end feature generator as an overlay on an existing CAD system. In this method FBMS can make use of the typical CAD facilities (like mouse interface, zooming, editing, etc.) offered by the CAD system and thus relieves the programmer or developer from building them from scratch. The constraint on this approach is that the CAD system must be available within the industries if CAD systems cannot support the runtime versions [21].

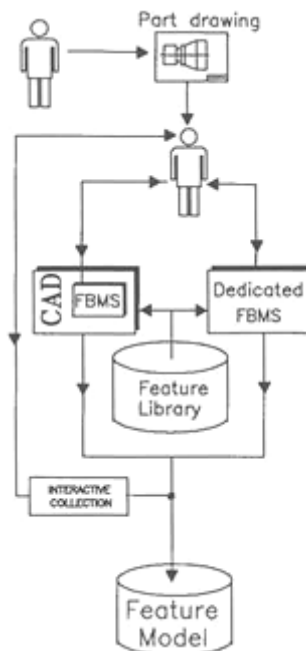


Figure 2.6 Feature-based modeling system (FBMS) [21]

2.9.3 Feature-Based Design Systems (FBDS)

Feature-based design facilitates the conceptual designing of mechanical parts in terms of their constituent parameterized features. Here the emphasis is on the conceptualization of a product during the design stage. Feature-based design essentially involves the feedback to the user that can be validity of the designs, design for manufacturability (DFM), and other related factors. Iterative loops between FBDS and the user until the user reaches a satisfactory design are crucial here. It can be considered as a pre-design method. Feature-based design is frequently used to describe post-design modeling methods. In fact, the characteristic of the feature-based design systems that distinguishes it from other systems is the origin of the model.

Other approaches (FRES and FBMS) start with an existing design (a drawing or a CAD data) to build the model, whereas this approach solely depends on the imagination and ability of the designer to use the features for designing the part. Hence, FBDS must provide the designer with the tool necessary to carry out the design analysis. These can also be developed using dedicated systems or general purpose CAD systems.

This offers the total solution because the part modeling and the design representation match with each other (both are in terms of features) and there will be no special need to carry out part modeling. The design representation can server the purpose of supplying the required data to the CAPP function [21].

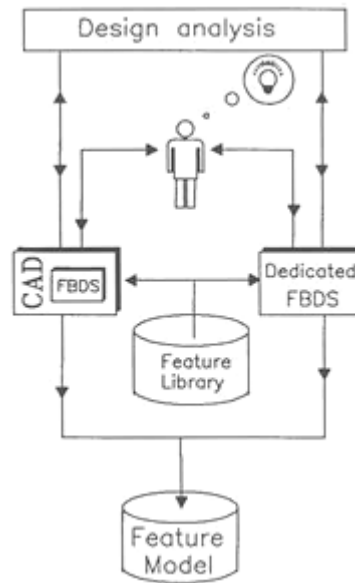


Figure 2.7 Feature-based design system (FBDS) [21]

2.10 CNC Part Programming

One of the first applications of computers in manufacturing was numerical control (NC). This is the technique of controlling machine tools with preordered and coded information. This technique was first developed in the 1950s to make contoured aircraft parts and since then it has spread into other industries, and today numerically controlled devices are used in all manner of industries.

The instructions of NC systems can be prepared either manually or with computer assistance. Manual programming starts with tool-path coordinates that are calculated from the part dimensions. It provides direct user control over NC instructions; however, with its numerous calculations it becomes tedious and easily prone to errors. The development of the modern controllers was enabled by the incorporation after about 1970 of a computer within the control itself, in so-called Computer Numerical Control (CNC).

In the days of NC (before CNC), one of the major limitations of this kind of equipment was the fact that the program had to be run from a tape. Holes punched in this tape allowed the program to be run from a tape reader of the NC machine

could understand. The program on the tape had to be perfect. The purpose of the tape reader is to read the tape into the memory of the control. That is, no longer is the program actually executed from the tape reader. This is one of the main differences between NC and CNC. The program stored in the CNC control can be easily modified. Some controls have floppy-disk drives and hard drives for program storage. The operator can easily call up the desired program and execute it any time.

In spite of the efforts of the manufacturers and the various professional organizations, the CNC industry has not so far been able to come up with a compatible coding standard. This was primarily because no national or international body has the power to regulate or enforce the uniform use of codes for manual programming. As a result, the same G Code may represent different functions on different systems. Because of this reason, post-processors are developed to compensate. A post-processor is a program that converts a toolpath, which contains all information to machine part to a NC program. The post-processor knows all the programmable codes (G, M, T, S, etc.), acceleration and deceleration characteristics, feeds and physical limitations of slide motions [25].

The name of the stored program in CNC is the part program, and the process of writing these is known as part programming. The instructions in a part program appear as a sequence of blocks containing commands to set machine operations, parameters, dimensional and speed data. Each command has associated identifying letter, and is generally identified by a number.

A part program can contain a number of separate programs, which together describe all the operations required to manufacture the part. Main program is the controlling program, i.e., the program first read, or accessed, when the entire part program sequence is run. This controlling program can then call a number of smaller programs into operation. These smaller programs, called Sub Programs, are generally used to perform repeated tasks, before returning back to the main program. Normally, controller operates according to one program. In this case the main program is also the part program.

Main programs are written using I.S.O. address codes, are listed below:

N- refers to block number,

G- refers to G code (preparatory function),

X- refers to absolute distance traveled by the slide tool in the X-axis direction,

U- refers to incremental distance traveled by the slide tool in the X-axis direction,

Z- refers to absolute distance traveled by the slide tool in the Z-axis direction,

W- refers to incremental distance traveled by the slide tool in the Z-axis direction,

F- refers to feed rate,

M- refers to M code (Miscellaneous function),

S- refers to spindle speed,

T- refers to tooling management.

Each block, or program line, contains addresses which appear in this order: N, G, X (or U), Z (or W), F, M, S, T. This order should be maintained throughout every block in the program, although individual blocks may not necessarily all these addresses [26].

2.11 Microsoft Distributed interNet Applications Architecture (DNA)

Microsoft Windows Distributed interNet Applications Architecture (Windows DNA) is the application development model for the Windows platform. Windows DNA specifies how to: develop robust, scalable, distributed applications using the Windows platform; extend existing data and external applications to support the Internet; and support a wide range of client devices maximizing the reach of an application. Windows DNA addresses requirements at all tiers of modern distributed applications: presentation, business logic, and data. Like the familiar PC environment, Windows DNA enables developers to build tightly integrated applications by accessing a rich set of application services in the Windows platform using a wide range of familiar tools. These services are exposed in a unified way through the Component Object Model (COM). Windows DNA provides customers with a roadmap for creating successful solutions that build on their existing

computing investments and will take them into the future. Using Windows DNA, any developer will be able to build or extend existing applications to combine the power and richness of the PC, the robustness of client/server computing, and the universal reach and global communications capabilities of the Internet [11].

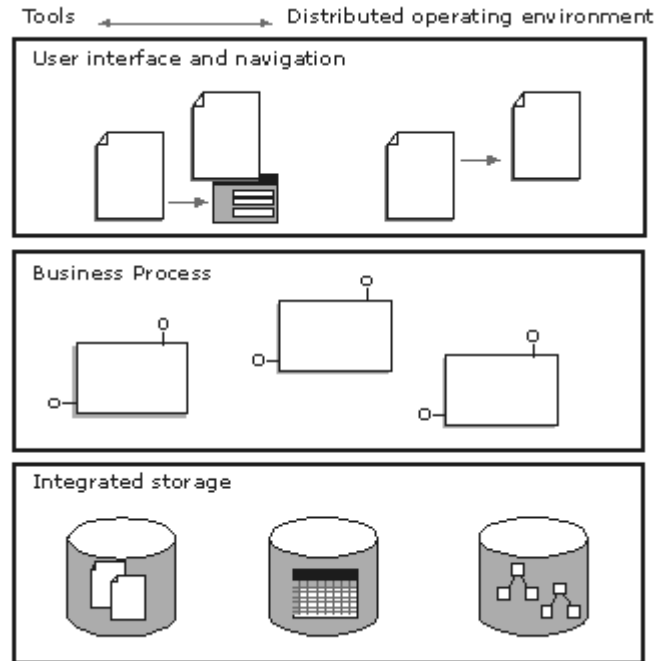


Figure 2.8 DNA architecture

2.12 Object Oriented Programming (OOP)

Object Oriented Programming is a software design methodology and programming paradigm that defines programs in terms of "classes of objects", objects being entities that combine both *state* (i.e., data) and *behavior* (i.e., procedures, or *methods*).

An object is an entity that embeds both data and the code that deals with it. It is created at run time, which requires memory and possibly other system resources and is then destroyed when it is no longer needed or when the application ends. A class is a portion of the program (a source code file, in Visual Basic) that defines the properties, methods and events of one or more objects that will be created during execution. In a sense, classes are design time-only entities, while objects are run time-only entities [27].

OOP should include both use of objects and some support of reusability including abstraction, inheritance. OOP expresses a computer program as a set of these objects, which encapsulate data with function and provide regulated communication between each other to perform tasks. This differs from traditional procedural languages in which data and procedures are separate. The objective of Object-oriented programming is to componentize software in order to make programs easier to write, maintain, reuse and prove correct.

Another way this is often expressed is that object-oriented programming encourages the programmer to think of programs primarily in terms of the data types, and secondarily on the operations ("methods") specific to those data types. Procedural languages encourage the programmer to think primarily in terms of procedures, and secondarily the data that those procedures operate on. Procedural programmers write functions, and then pass data to them. Object-oriented programmers define objects with data and methods, and then send messages to the objects telling them to perform those methods on themselves.

There is some disagreement about exactly which features of a programming method or language qualify as "object-oriented", but there is a general consensus that the following features are most important.

- **Abstraction:** Each object in the system serves as a model of an abstract "actor" that can perform work, report on and change its state, and "communicate" with other objects in the system, without revealing *how* these features are implemented. Processes, functions or methods may also be so abstracted, and when they are, a variety of techniques are required to extend an abstraction.
- **Encapsulation:** Also called "information hiding", this ensures that objects cannot change the internal state of other objects in unexpected ways; only the object's own internal methods are allowed to access its state. Each type of object exposes an *interface* to other objects that specifies how other

objects may interact with it. Some languages relax this, allowing some direct access to object internals in a controlled way, and limiting the degree of abstraction.

- **Polymorphism:** References to and collections of objects may contain objects of different types, and invoking a behavior on a reference will produce the correct behavior for the actual type of the referent. When it occurs at "run time", this latter feature is called *late binding* or *dynamic binding*. Some languages provide more static ("compile time") means of polymorphism such as C++ templates and operator overloading.
- **Inheritance:** Organizes and facilitates polymorphism and encapsulation by permitting objects to be defined and created that are specialized types of already-existing objects - these can share (and extend) their behavior without having to re-implement that behavior. This is typically done by grouping objects into *classes*, and classes into *trees* or *lattices* reflecting common behavior.

Object-oriented programming "took off" as the dominant programming methodology during the mid-1980s, largely due to the influence of C++, an extension of the C programming language. Its dominance was further strengthened by the rising popularity of graphical user interfaces, for which object-oriented programming is particularly well suited [28].

2.13 Unified Modeling Language (UML)

Unified Modeling Language is a general-purpose visual modeling language that is used to specify, visualize, construct, and document the artifacts of a software system. It captures decisions and understanding about the systems that must be constructed. It used to understand, design, browse, configure, maintain and control information about such systems. UML includes semantic concepts, notation and guidelines. It has static, dynamic, environmental, and organizational parts.

In UML, there are three main modeling viewpoints, namely: “use case” models, “static” models and “dynamic” models. Use case models describe system requirements from user viewpoints. Static models are essentially class diagrams that describe system elements and their relationships (including generalization, aggregation and association relationships) [29].

2.13.1 UML Use Cases and Use Case Diagrams

Use cases define generic processes the system must be able to handle. The building blocks of use case models are use cases, actors and the system modeled. The use case, represented graphically as a named oval, specifies the functionality provided by the system or a task that has to be done with support from the system. An actor, usually shown as a stick person, is not part of the system, but an external entity that must interact with the system. An actor is a type (a class), not an instance. Each type of actor represents a role, not an individual user of the system. Actors communicate with the system by sending and receiving messages. When an actor sends a message to a system this will initiate a use case [29].

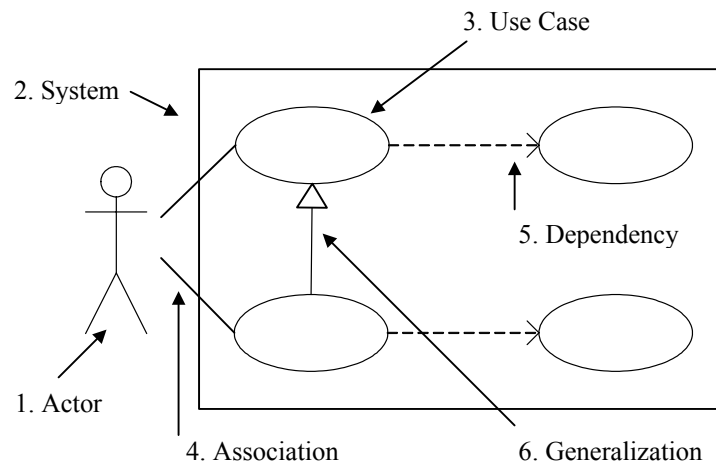


Figure 2.9 Elements of Use Case diagram [29]

2.13.2 UML class diagrams and static modeling

The term “class” is a descriptor used in UML notation to refer to a set of objects with similar data structure, behavior and relationships. A class diagram can be used to provide a static view of a system in terms of its object classes and the relationships among those classes. However, class diagrams do not encode temporal information. Four kinds of UML modeling construct (association, composition, generalization and dependency) are used to describe static relationships within class diagrams [29].

2.13.3 Dynamic Modeling

Generally, system objects need to communicate with other system objects and objects external to that system. In UML notation, object interactions are described using a dynamic model. Modified system behavior is normally initiated following the transmission of a message from one object to another. In UML, modeling constructs are provided to describe the four types of message namely synchronous, synchronous with immediate return, asynchronous and simple. UML also provides modeling constructs to construct four types of dynamic diagram, namely state diagrams, sequence diagrams, collaboration diagrams and activity diagrams. State diagrams and activity diagrams can be used to encode structural descriptions, whereas sequence diagrams and collaboration diagrams are designed to describe how behavioral descriptions are executed [29].

2.13.4 State Diagram

State and change of state will be encoded by values attributed to an object. State diagrams are typically deployed to complement the description of a class. A state diagram can show all possible states that can be reached by objects of the class, and which events trigger/cause state changes. An event can be encoded by an object initiating a message [29].

2.13.5 Activity Diagram

Activity diagrams are used to model flows of activities, such as activities comprising a procedure. In UML, it is assumed that states will be changed by activities, while transitions will be related to actions. Consequently, activity diagrams are inherently linked to state diagrams. In an activity diagram, events are generally referred to as transitions. Unlike state diagrams, focus here is on describing events that occur within a single object as it responds to message instances. An activity diagram can be used to model an entire business process. Further, it can provide a high level view of what is going on inside a use case. Modeled elements in an activity diagram are activities, transitions, synchronization bars, decision diamonds and start and stop markers [29].

2.13.6 Sequence Diagram

A sequence diagram is used to show how objects and actors take part in a collaboration. It defines and illustrates the sequence in which objects pass messages to each other. Thereby, the sequence diagram focuses on message sequences over time. A sequence diagram has two dimensions, namely: a vertical dimension used to represent time and a horizontal dimension used to represent aspects involved in a collaboration. The modeling constructs of a sequence diagram comprise objects, links and messages [29].

2.13.7 Collaboration Diagram

In UML notation, collaboration defines relationships between a set of objects in a particular context. Collaboration diagrams can be used to show the sequence in which events occur by describing the structural organization of objects that exchange messages. Here message arrows are drawn between objects to define and illustrate message flows between objects. Modeled elements of a sequence diagram are: objects, links and messages. Collaboration diagrams and sequence diagrams

encode similar information. Which is best to deploy will depend upon what aspect of a given interaction the user needs to focus on [29].

2.14 Overview of the Studies on CAD/CAM, CAPP

Some of the existing studies in the literature are written below:

A rule-based approach to sequence machining operations of rotational components was developed by Saygin and Eskicioğlu [30]. For this purpose the IRPPS (Interpreted Rule-Based Process Planning System) was developed. It was using a set of rules to extract and interpret the geometric features of a rotational part from the AutoCAD drawing of the part. The feature recognition module of the system can be used as a design and drafting tool for rotational components. Rules and facts are grouped in sets and each set of rule is represented by a rule name in order to obtain a knowledge base, which can be modified easily without making any changes in the inference part of the system. The machining area between the finished product and blank material is represented by cutting planes and rules are applied to remove those cutting planes in an order so that the rotational part is obtained. The system can also select the operations and cutting tools needed to turn the part and the sequence of the operations. The system output is a process plan including the sequencing of machining operations for external, internal, superimposed features as well as the corresponding cutting tools and machinability data. The selection of inserts and tool holders makes it possible to use the generated process plans on a CNC machine tool.

Mestçi, Sevinç and Akkurt [31] have worked on a CAD/CAM package. They have developed a computer integrated design and manufacturing program for rotational parts to generate NC cutting path files directly from design geometry. Blank material data is chosen from the present ten materials and they are assumed to be cylindrical in shape. The program takes feed, cutting speed and tool life into account while generating the NC part program. The program generates a general

purpose NC code for parts. In order to use this code in any CNC machine the code should be modified according to the machine tool.

Arslan and Söylemez [32] have worked on the extraction of required data to prepare part program by using variant approach method. It was taking the product data from a database in the form of DXF format for the cylindrical parts and tried to obtain output as NC code for the part. Part definition is done by AutoCAPP-TOR automatically by reading the DXF file. Rotational parts are considered as the composition of simple geometric parts and divided into segments starting from the left.

Filiz, Kayacan and Dereli [33] have developed a modeling system called as ROMOD to recognize rotational parts. The finished parts are represented as a set of geometrical entities such as lines, arcs, etc. and it is drawn by using the computer aided drafting module integrated into the system. After the part recognition is completed it is stored in a CAD database to be used in a computer aided process planning system. A computer program was also developed to generate CNC part programs.

Talışık and Eskicioğlu [34] have developed a process planning package for rotational parts. In this application, the Interpreted Rule-based Process Planning System (IRPPS) which was developed by Saygın [30], was recoded using C++ programming language. A feature description module was added to the system. The existing tool selection module in IRPPS was improved by adding a new selection criteria and enhancing user interaction. Determination of cutting planes for internal features was not considered in the IRPPS, but this deficiency was removed in their system by using the logic that was applied to external features in IRPPS. Moreover, there exists a G Code generation module in the package.

A knowledge-based approach to generative process planning was presented at the CASA/SME Autofact Conference, by Phillips and Mouleeswaran [35]. In this research, AI concepts and knowledge-based reasoning techniques were used to

develop a methodology for combining a CAD system with manufacturing decision logic, so that process plans can be generated with little or no manual intervention. The developed system for rotational parts, namely PROPLAN, derives parts geometry and other related information from a CAD database in the form of a descriptive language, and produces a detailed process plan to manufacture the parts. It uses “top-down” approach.

The knowledge base of PROPLAN consists of production rules on various aspects such as selection of machine tools, selection of machine tools, selection of machining parameters, etc. These rules are of the form IF-THEN structure. Its knowledge base is capable of

- Selection of process/operation
- Selection of machine tool
- Selection of tools and accessories
- Selection of machining parameters like speed, feed depth of cut
- Selection of coolant

In PROPLAN, the representation of the problem is a two-dimensional graph that shows the area to be removed in order to develop the process plan of the part. Real world three dimensional (3D) graphics are viewed as operators that remove two dimensional (2D) primitive shapes from the grid. For example, decreasing the diameter of raw material in real world (2D). This method was found to be very efficient. After the shapes to be removed are determined, some heuristics* are used to develop the solution path.

Another generative CAPP system for rotational parts, called ROUND, was presented by Houten [36]. It is developed in the Laboratory of Production Engineering at Twente University of Technology in Netherlands. The program starts with the execution of the input module. If the blank material is pre-shaped or a casting, the user defines the dimensions of it, else a module determines the

* Heuristic: A method of learning involves discovery and problem solving, using reasoning and past experience. (Collins COBUILD Dictionary, 2001)

dimensions due to the feasible clamping method. In order to select the machining methods for roughing operations, the volume to be removed is divided into a number of areas, to each of which a specific machining operation is assigned. Automatic machine tool selection is not available so the operator should select the machine tools.

KAPLAN, which is another knowledge-based approach to process planning of rotational parts, was developed by Giusti et al. [37]. The program structure is based on knowledge-based techniques and the knowledge required for process plan generation is represented by IF-THEN rules, easily adaptable to every workshop environment by means of a user interface. Integration between CAD and CAPP, obtained by means of design module, has allowed the management of data flow in a full automatic way up to the production plan with all the needed information. Data reliability and rapid process planning is achieved by this integration. KAPLAN can be considered a valid and new approach to the CAPP for rotational parts. This is resulted from that, in KAPLAN, all the possible plans for a mechanical part is generated and best one is selected according to the selection criteria stored in the database. The strategy to generate the process plans is based on the analysis of all the surfaces, which can be clamped and on the processing of the consequent group of surfaces to be machined.

An expert system was developed by Boer et al. [38]. This semi-generative CAPP/CAM system is for a multi-turret NC automatic turning-milling center. Due to the system architecture, at first, the definition of the operations and of the associated tools necessary to obtain the finished part starting from the bar is given by the user, both in terms of geometry and precision. Next, the technology module calculates for each operation itself. The following expert system module is based on rules that have been introduced according to the experience of the technicians. This module is capable of interacting with the user. Next scheduling module prepares the operations scheduling in respect to the given constraints. At last, the part program generation module generates the tool paths.

An integrated process planning and monitoring system for turning operations was developed by K. K. Cho et al. [39]. This system was aimed for small and medium sized machinery industry. AutoCAD was employed as CAD program. CLIPS (C Language Integrated Production System) was used as an expert system shell for knowledge based process planning. dBASE III was used as a database management software. The developed integrated system consists of automatic process planning system including NC programming, process data management system and process monitoring system.

Another integrated knowledge-based machining system for rotational parts were developed by Yeo et al. [40]. This research models the knowledge of an experienced process planner and NC programmer by combining expert system techniques. This system is also aimed for small and medium-sized machinery industry. The software tool GOLDWORKS II with COMMON LISP is used as programming platform. The developed system was generative because in process planning, the information generated in various stages in can be modified by the user.

A CAPP expert system for rotational components was reported in 1997 by M.A. Younis and M. A. Abdel Wahab [41]. Their involved effort was limited to the manufacturing of rotational components using metal cutting operations and generative process planning is chosen over variant -group technology- process planning. Expert systems differ from databases in that their knowledge is not simply readable as data, but can also be used for the solution of different types of problems. Knowledge is thus distinguished from data in that it is integrated with instructions for its use. The division of the knowledge into as many fragments as possible makes a knowledge base more modeler and therefore easy to modify. The basic knowledge representation types are:

- Rules, which consists of preconditions and actions (if X then Y)
- Facts, which describe the variables manipulated by the rules.

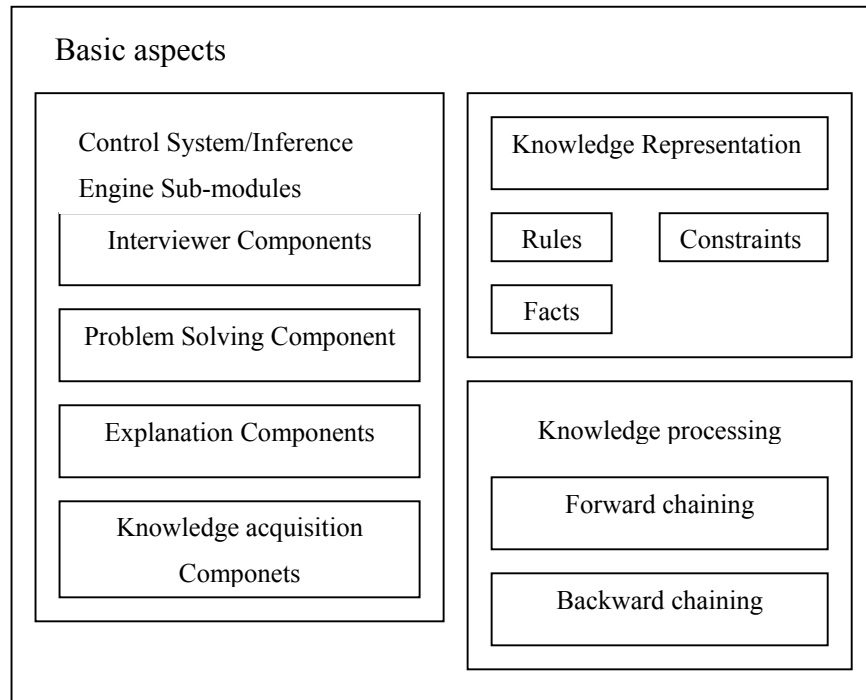


Figure 2.10 Expert system basic aspects [41]

2.15 Overview of the Studies on Internet-Based Design and Manufacturing

An example study was made in University of California Berkley called CyberCut Project [42,43]. The project concerns with facilitating distributed design and fabrication processes with the evolution of CAD/CAM tools. Their effort is to develop a fabrication agent service and a web-based design tool for manufacturing parts over the Internet. The fabrication agent is build on an open-architecture machining system for experiments on and machining of 2.5D and freeform parts over Internet. Java is chosen as programming development tool. The service allows client designers on the Internet to create mechanical components and submit appropriate files to the server at Berkeley for process planning and fabrication on an open-architecture CNC machine tool. Manufacturing advisory service (MAS) and process planning modules are also included in their project. The MAS helps designer unfamiliar with manufacturing processes to select an appropriate method for their part. The MAS analyzes a conceptual description of the part and returns a ranked set of processes. In process planning, software modules called the macro

planner, micro planner, and tool path planner are responsible for automatically producing machining instructions.

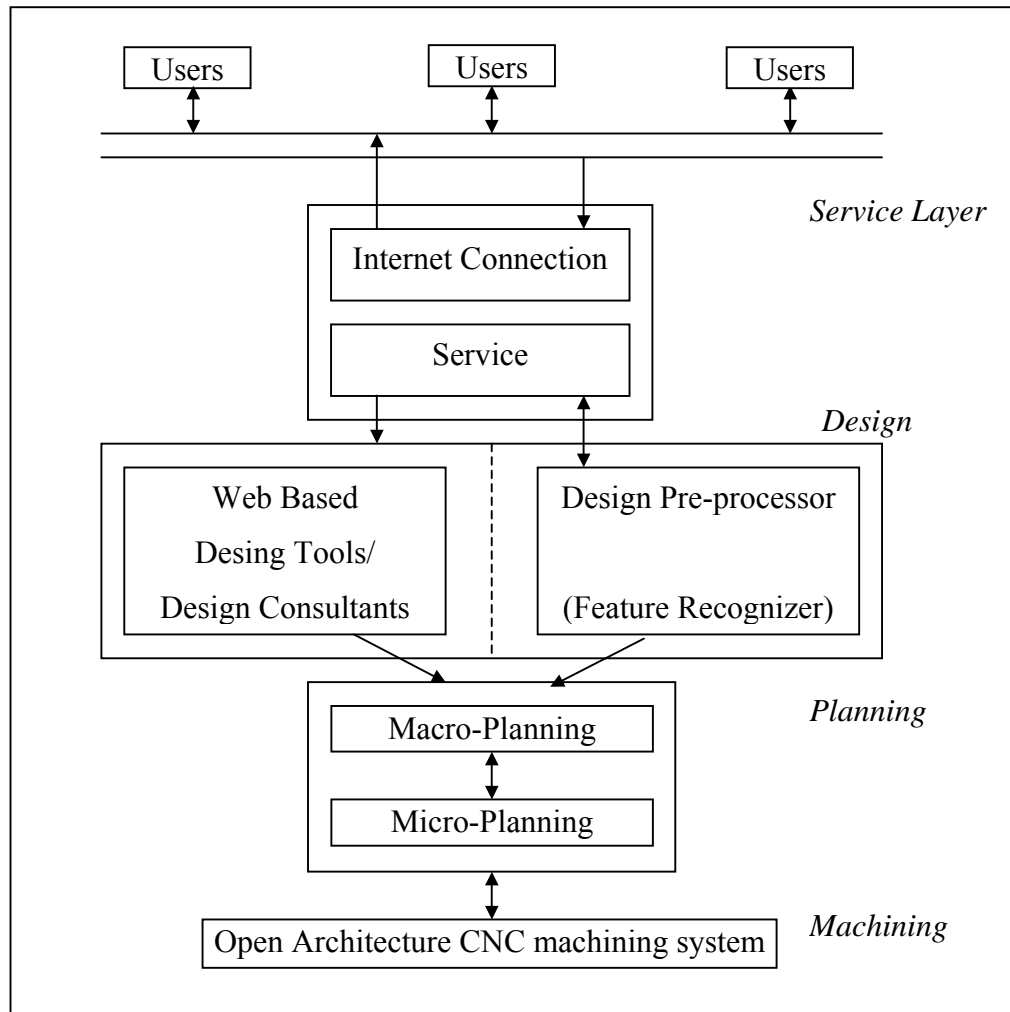


Figure 2.11 Architecture of the CyberCut Service [43]

A web-based cutting tool and parameter database was developed by Z. Adamczyk and Krzysztof Kociol [44]. PAFANA and BAILDONIT tool and insert catalogues were used in their system. They published the HTML documents containing tools and inserts data of the catalogues, including hyperlinks among them, which define the way of searching the document structure.

G.Q. Huang, S. W. Lee and K. L. Mak. [45] proposed a web-based product and process data modeling in concurrent “design for X”. Their purpose was to create DFX shell onto the Internet/Intranets using the web technology. Their focus was on

the pragmatic techniques appropriate for modeling not only product and process data but also their interactions and consumption. They made use of the Microsoft ActiveX technology and in their web application.

A distributed and collaborative concurrent product design system through the Internet was suggested by Quang, Zhang and C. Nee [46]. Their work allows participants located anywhere in the world to collaborate in real-time. The general architecture of the system is a client/server architecture in which one server supports many clients through the Internet. The server supplies CAD geometry and engineering information and clients share this information through the server. For real-time codesign based on traditional commercial CAD software to a multi-location application through the Internet.

CHAPTER 3

PROPOSED SYSTEM STRUCTURE

In this chapter, the overall structure of the Project WebTURN is described in a comprehensive manner. Besides that, complementary figures and tables are provided.

In METU-CIM a long work, called AutoTURN has been done by four people. It was a CAD/CAM software package for rotational parts. It was making use of AutoCAD at the background and running on Windows operating systems. İlik [47] was the first who started the work and he made the first feature modeler, then Öztürk [48] has included the machinability and tool databases and capability of automatic tool selection then Başsoy [19] has added the process planning module and finally Susuzlu [49] has made the NC verification of the part program generated by the AutoTURN.

This study focuses on a different working environment, which is the WWW (World Wide Web). Hence, the distributed design and manufacturing forms the objective of this work. Web programming differs from stand-alone softwares in terms of the development tools and methodology.

The effort involves enabling the distant users to make 2D wireframe geometric model of rotational parts and to see the concurrent process plan and part program of the designs required for manufacturing.

3.1 Design Features Description and Product Data Representation

WebTURN proposes a feature-based representation and modeling methodology for rotational parts based on a build feature taxonomy. The taxonomy of the rotational design features is shown in the Table 3.1.

In CAD/CAM systems, it is important to use a consistent format in documenting the product geometric data within the system itself. That is why, a consistent product data representation format for the definition of feature taxonomy is developed to store and share product data. Sample product data representations are shown in Figures 3.1 and Figure 3.2.

At the header of the part data file, name and surname of the part owner is written. The part creation date is typed at the second line. The knowledge of features is written from next line till the end of file. The feature data starts with the class of the feature that indicates which primary class the feature belongs to which are turn, taper, convex&concave or internal features. Then, type of the feature is written, i.e. rightendconvexfillet, leftendchamfer etc. If there is a defined surface roughness, the value of it is written beside the type of feature. After that, all dimensional specifications of the feature, which are diameter, length and tolerances of them and if there exists the details of all sub-features are written.

3.2 Geometric Part Design

Application's design tool permits online users to design rotational parts by inserting the related features onto sketching window. Part features are attached end by end from left to right until the objective part geometry is constituted. There are five major features in the system, which are turn feature, taper feature, convex feature, concave feature and internal feature. Furthermore, some sub-features like fillets, chamfers, grooves and threads can also be added to those primary features. The design tool also creates the product data file and transports it to the active server.

Table 3.1 Taxonomy of rotational features

No.	Design feature (parameter1, parameter2, ...)
	TURN:
1	straightcylinder (D, \pm tol.s, L, \pm tol.s, SF)
2	rightendconvexfillet (D, \pm tol.s, L, \pm tol.s, R1 SF)
3	rightendconcavefillet (D, \pm tol.s, L, \pm tol.s, R1 SF)
4	leftendconvexfillet (D, \pm tol.s, L, \pm tol.s, R2, SF)
5	leftendconcavefillet (D, \pm tol.s, L, \pm tol.s, R2, SF)
6	rightendconvexfillet_ leftendconvexfillet (D, \pm tol.s, L, \pm tol.s, R1,R2, SF)
7	rightendconvexfillet_ leftendconcavefillet (D, \pm tol.s, L, \pm tol.s, R1,R2, SF)
8	rightendconcavefillet_ leftendconvexfillet (D, \pm tol.s, L, \pm tol.s, R1,R2, SF)
9	rightendchamfer (D, \pm tol.s, L, \pm tol.s, setbacksize1, α 1, SF)
10	leftendchamfer (D, \pm tol.s, L, \pm tol.s, setbacksize2, α 2, SF)
11	rightendchamfer_ leftendchamfer (D, \pm tol.s, L, \pm tol.s, setbacksize1, α 1, setbacksize2, α 2, SF)
12	leftendconvexfillet_ rightendchamfer (D, \pm tol.s, L, \pm tol.s, setbacksize1, α 1, R2, SF)
13	leftendconcavefillet_ rightendchamfer (D, \pm tol.s, L, \pm tol.s, setbacksize1, α 1, R2, SF)
14	rightendconvexfillet_ leftendchamfer (D, \pm tol.s, L, \pm tol.s, R1, setbacksize2, α 2, SF)
15	rightendconcavefillet_ leftendchamfer (D, \pm tol.s, L, \pm tol.s, R1, setbacksize2, α 2, SF)
16	thread (D, \pm tol.s, L, \pm tol.s, x1, x2, pitch)
17	rightendchamfer_ thread (D, \pm tol.s, L, \pm tol.s, setbacksize1, α 1, x1, x2, pitch)
18	rectangulargroove (D, \pm tol.s, L, \pm tol.s, gwidth, gdepth, greference)
19	roundgroove (D, \pm tol.s, L, \pm tol.s, gwidth, gdepth, greference)
20	leftendconcavefillet_ rectangulargroove (D, \pm tol.s, L, \pm tol.s, R2, gwidth, gdepth, greference, SF)
21	rightendchamfer_ rectangulargroove (D, \pm tol.s, L, \pm tol.s, setbacksize1, α 1, gwidth, gdepth, greference, SF)
22	leftendchamfer_ rectangulargroove (D, \pm tol.s, L, \pm tol.s, setbacksize2, α 2, gwidth, gdepth, greference, SF)
	TAPER:
1	convergingtaper (D1, \pm tol.s, D2, \pm tol.s, L, \pm tol.s, SF)
2	divergingtaper (D1, \pm tol.s, D2, \pm tol.s, L, \pm tol.s, SF)
3	convergingtaper_ rightendchamfer (D1, \pm tol.s, D2, \pm tol.s, L, \pm tol.s, setbacksize1, α 1, SF)
4	convergingtaper_ rightendchamfer_ leftendchamfer (D1, \pm tol.s, D2, \pm tol.s, L, \pm tol.s, SB1, α 1, SB2, α 2, SF)
5	divergingtaper_ rightendchamfer (D1, \pm tol.s, D2, \pm tol.s, L, \pm tol.s, setbacksize1, α 1, SF)
6	divergingtaper_ rightendchamfer_ leftendchamfer (D1, \pm tol.s, D2, \pm tol.s, L, \pm tol.s, SB1, α 1, SB2, α 2, SF)
	CONVEX&CONCAVE:
1	Convex (D1, \pm tol.s, D2, \pm tol.s, L, \pm tol.s, R, SF)
2	Concave (D1, \pm tol.s, D2, \pm tol.s, L, \pm tol.s, R, SF)
	INTERNAL
1	Drill (D, L, bottomcondition)

R: radius, R1: right fillet radius, R2: left fillet radius
D: diameter, D1: first diameter, D2: second diameter
 θ 1: right chamfer angle, θ 2: left chamfer angle
x1: thread starting point, x2: thread ending point
SF: surface roughness, tol.s: tolerances

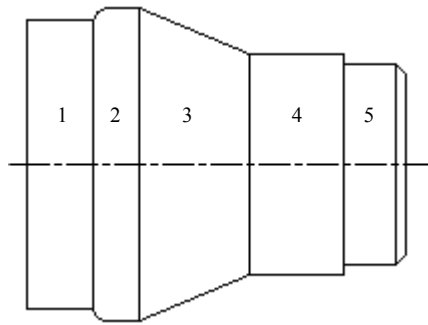


Figure 3.1 A sample part drawing, comprised of five external features and its data file.

```
"Owner", "Name Surname"
"Creation Date", "Date"
"Feature", "Turn"
"Type", "StraightCylinder", SR
"D", D, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
"Feature", "Turn"
"Type", "LHConvexFillet", SR
"D", D, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
"LHConvexFillet", SR
"Feature", "Taper"
"Type", "Converging_Taper", SR
"D1", D1, +tolerance, -tolerance
"D2", D2, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
"Feature", "Turn"
"Type", "StraightCylinder", SR
"D", D, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
"Feature", "Turn"
"Type", "RChamfer", SR
"D", D, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
"RChamfer", setbacksize1, 01
```

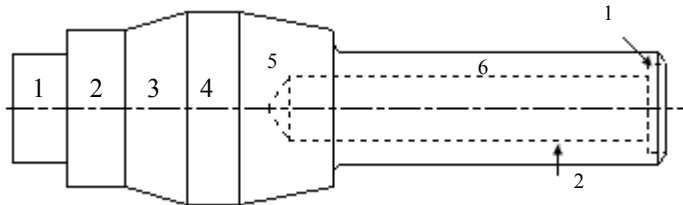


Figure 3.2 A sample part drawing, comprised of six external and two internal features and its data file.

```
"Owner", "Name Surname"
"Creation Date", "Date"
"Feature", "Turn"
"Type", "StraightCylinder", SR
"D", D, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
"Feature", "Turn"
"Type", "StraightCylinder", SR
"D", D, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
"Feature", "Taper"
"Type", "Diverging_Taper", SR
"D1", D1, +tolerance, -tolerance
"D2", D2, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
"Feature", "Turn"
"Type", "StraightCylinder", SR
"D", D, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
```

→

↓

```
"Feature", "Taper"
"Type", "Converging_Taper", SR
"D1", D1, +tolerance, -tolerance
"D2", D2, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
"Feature", "Turn"
"Type", "LHConcaveFillet_RChamfer", SR
"D", D, +tolerance, -tolerance
"L", L, +tolerance, -tolerance
"RChamfer", setbacksize1, 01
"LHConcaveFillet", R2
"Feature", "Internal"
"Type", "FlatEndDrill"
"D", D
"L", L
"Feature", "Internal"
"Type", "ConicalEndDrill"
"D", D
"L", L
```

3.3 Interpretation of Product Data File

When the part data file is received from a user, reading of the file starts. The part data file is read line by line and defined feature or features are recognized sequentially. At the end of the feature recognition procedure, system learns the all feature composition and feature specification of the input design. After the features are recognized, the largest diameter is searched and founded among the features. Total length of the part is calculated by summing the constituting length of the features. Part is divided into two zones from the feature that possesses the largest sectional diameter. This feature and the feature sequence number are considered to be reference feature and reference feature index number, respectively.

3.4 Process Planning Procedure

The followed procedure in process planning is the backward planning since it makes planning easier. From most right feature to reference feature, machining pockets are identified and filled. Same method is performed from most left feature to reference feature to finally reach blank shape that is the billet size. The procedure on a sample part is shown is Figure 3.3.

The values of total part length and maximum diameter are passed to the process planning stage. Initially, a suitable billet size should be determined. The billet diameter is chosen from the possible standard billet size diameters listed to the user. Also the minimum possible billet length is displayed. User can change this value. If the entered value is validated by the system that is, if the input value is between the maximum possible turned length and the minimum possible billet length values, then it is used in the process planning. Then the coordinates of the nodal points are calculated according to the coordinate system of the CNC. The nodal points of a feature is defined as the intersection points of the major lines and arcs which forms the feature. The nodal points of a sample drawing are shown in Figure 3.4. While the machining pockets are identified, associated cutting operations are also decided.

Required number of cutting passes, coordinates of all cutting paths are calculated.

For each machining pocket, the machining time and power are calculated.

Machining process sequence on the lathe is:

1. Facing the right end of the external profile,
2. Overall horizontal rough cut up to the maximum diameter,
3. Rough right cut of external profile,
4. Rough left cut of external profile,
5. Finish right cut of external profile,
6. Finish left cut of external profile,
7. Grooving,
8. Threading,
9. Drilling,
10. Parting-off.

3.5 Machining Calculations

The detailed machining instructions required to manufacture the part on the Denford Mirac CNC turning machine, are prepared sequentially in process planning. In addition to machining listing, required powers, approximate machining times and the cutting parameters, which are feed rate, depth of cut and the cutting speed are also tabulated for individual cuts.

In cylindrical turning operations, the cutting speed at the tool corner is calculated by $\pi d_m n_w$, where n_w is the rotational speed of the workpiece and d_m is the diameter of the machined surface. The maximum value of the cutting speed is given by $\pi d_w n_w$, where d_w is the diameter of the work surface. Thus, the average or mean cutting speed v_{av} is estimated by

$$v_{av} = \frac{\pi n_w (d_w + d_m)}{2} \quad (1)$$

The metal-removal rate Z_w is the product of the mean cutting speed and the cross sectional area of the uncut chip A_c . Hence,

$$Z_w = A_c v_{av} = f a_p v_{av} \quad (2)$$

Machining times and metal-removal rates for different operations are:

Table 3.2 Equations for cutting operations

Operation	Machining time (t_m)	Metal-removal rate (Z_w)
Turning	$\frac{l_w}{f n_w}$	$\pi f a_p n_w (d_m + a_p)$
Facing	$\frac{d_m}{2 f n_w}$	$\pi f a_p n_w d_m$
Parting	$\frac{d_m}{2 f n_w}$	$\pi f a_p n_w d_m$
Drilling	$\frac{l_w}{f n_t}$	$(\pi f d_m^2 n_w) / 4$

The required machine power for each operation is calculated. The power is calculated by the formula:

$$\text{Power} = (P_s Z_w) / \eta_m \quad (3)$$

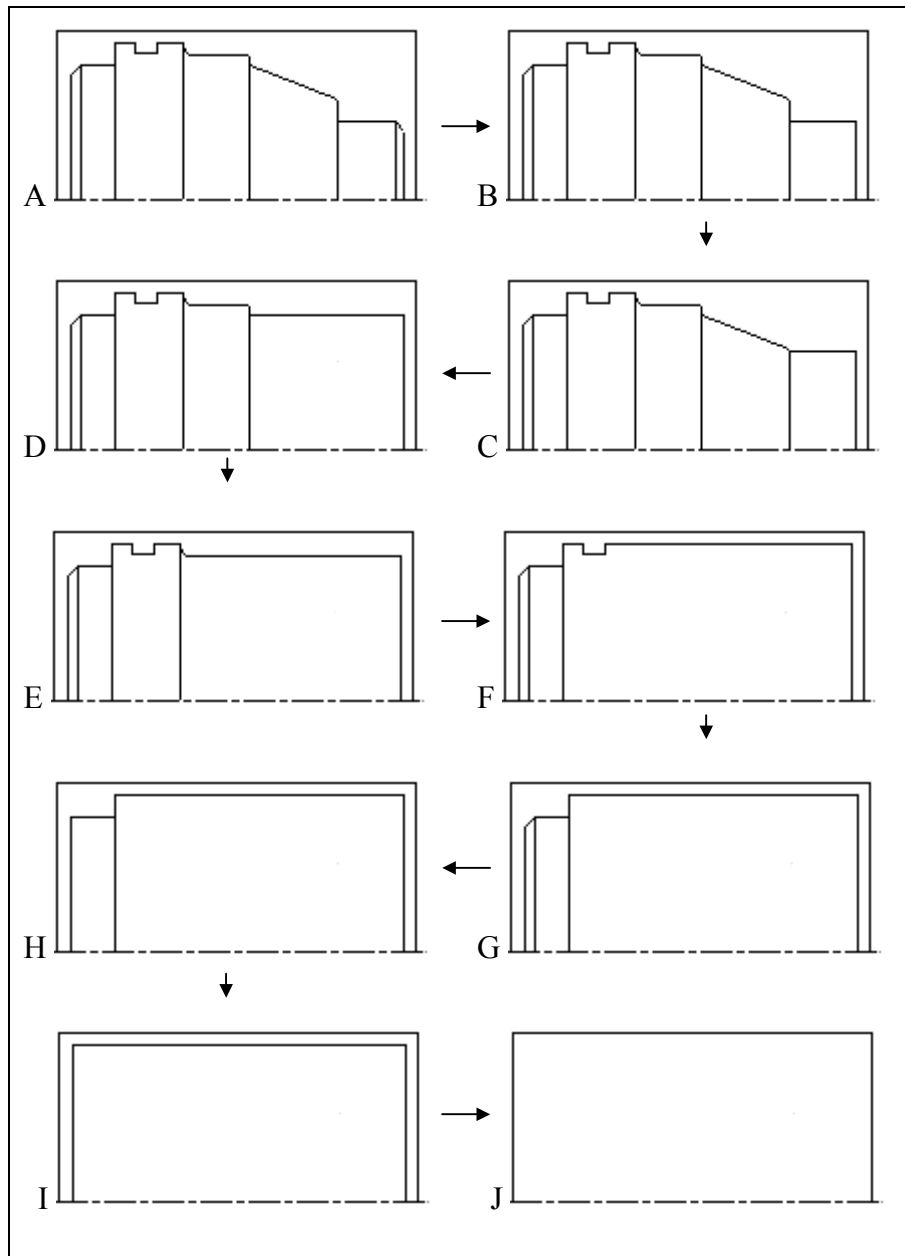


Figure 3.3 Steps in pocket identification procedure

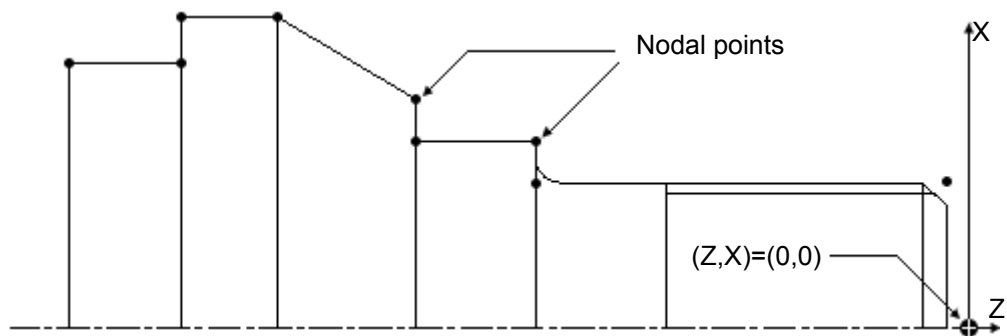


Figure 3.4 Nodal points on a sample part drawing

3.6 Part Programming

After the generation of operation sequence in process planning stage, the corresponding part program is possibly created by the post processor module of the application. A part program is a list of coded instructions, which describes how the designed component, or part, will be manufactured. These coded instructions are called data, a series of letters and numbers. The part program includes all the geometrical data to perform the required machine functions and movements to manufacture the part.

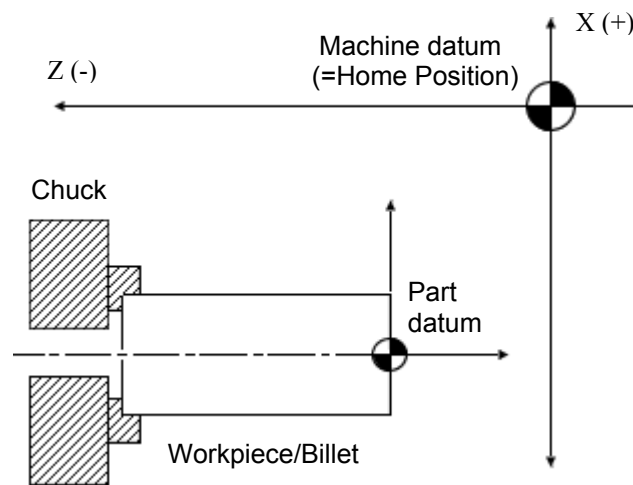


Figure 3.5 Working envelope of the CNC lathe

The Denford CNC lathe can plot coordinate movements using two different methods:

3.6.1 Machine Coordinates: Machine coordinates are the coordinates plotted with respect to the fixed machine datum position. The coordinate display always shows the true position of the machine. The machine datum position is set by machine tool manufacturer. It is impossible to change machine datum position since it defines the physical movement capability of the machine. When the machine coordinates are used the machine datum (home position) is displayed with the position $X=0$, $Z=0$.

3.6.2 Workpiece Coordinates: It is the coordinate value plotted with respect to the programmed workpiece datum. Coordinate display shows the position of the

machine when offsets are used. Workpiece datum is set by the operator. It is the location where the machining should begin. Offsets temporarily shift the entire coordinate based grid system of the machine. When workpiece coordinates are used billet/workpiece datum is displayed with the position $X=0, Z=0$.

There are two alternative programming methods for a CNC lathe, which are either radius programming or diameter programming. Diameter programming means all the X values programmed are relative to overall diameter of the workpiece. This is the most common programming way for a lathe because it is easier to understand where the tool is and which diameter it is obtained. Radius programming means that all X values are coded relative to the radius of the part. Radius programming is less common but it can be used where the dimensions of an engineering drawing is represented by radius values.

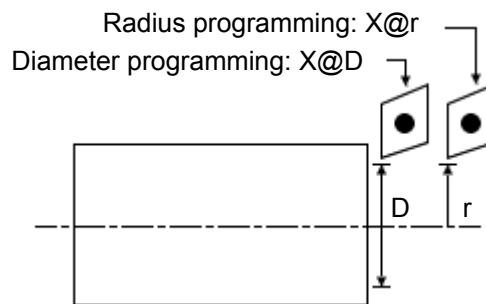


Figure 3.6 Diameter and radius programming

3.7 Machine Tool Offsets

When a CNC program is written, all coordinates used for describing the size of the part are indicated relative to the zero reference called machine datum. In G Code generation, the part datum is positioned on to the center of right end of the billet. By positioning like that, any negative Z values will indicate that the tool is cutting into the billet whereas, any positive Z values will indicate that the tool is away from the billet and doesn't make cutting. The assigned X coordinates point the diameter or the radius of the cutting path.

3.8 Basic G and M Codes in Part Programming

Some of the basic G and M codes used in part programming are described below:

3.8.1 G Codes (Preparatory Functions)

- **G00 (Rapid positioning/traverse):** It executes a non-cutting movement at a rapid feed rate to a specific coordinate point in the working area (absolute co-ordinate movement) or when a certain distance from a previously stated position (incremental co-ordinate movement) is programmed. G00 command is written in G00 X___ Z ___, format where G00 is the rapid positioning code, X___, Z ___ are the X and Z-axis coordinate values.

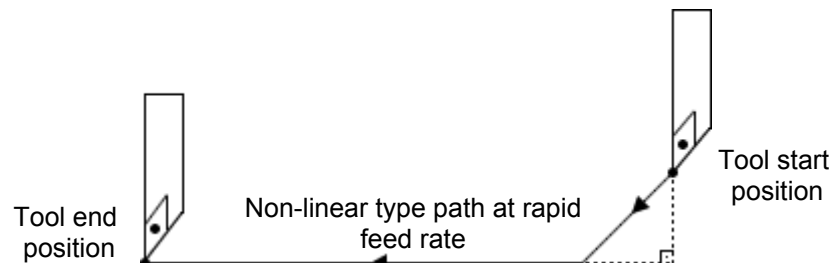


Figure 3.7 Movement of G00

- **G01 (Linear interpolation):** It executes a cutting movement following a straight line at a set feed rate. A G01 command is written in the following format: G01 X(or U)___ Z(or W)____. The feed rate value assigned to the G01 command is the actual feed rate along the proposed tool path, not the feed rate of axis or slide.

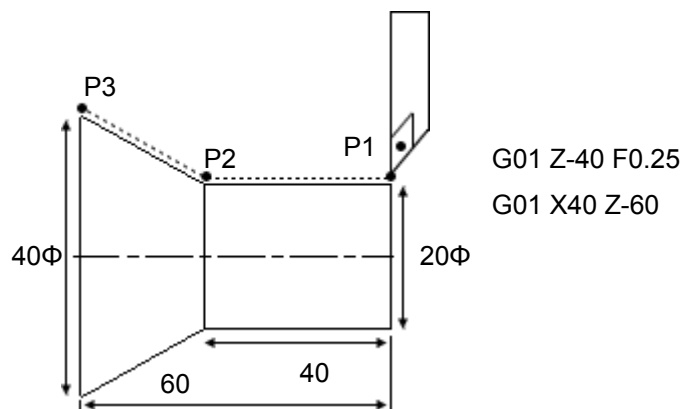


Figure 3.8 Movement of G01

- **G02 (Circular Interpolation-CW):** It executes a cutting movement following a clockwise circular path at set feed rate.

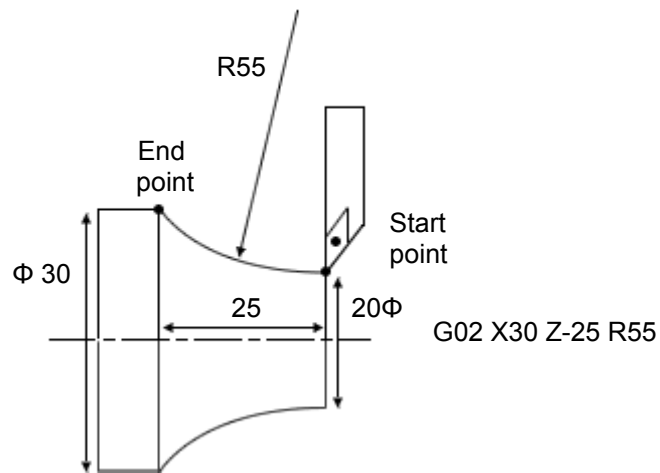


Figure 3.9 Movement of G02

- **G03 (Circular Interpolation-CCW):** It executes a cutting movement following a counter clockwise circular path at set feed rate.

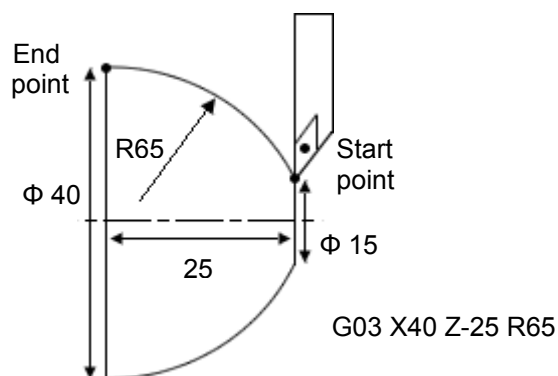


Figure 3.10 Movement of G03

- **G21:** Metric data input
- **G28:** Reference point return
- **G50:** Maximum spindle speed setting
- **G92 (Thread cutting cycle):** The G92 command performs a one pass threading cycle. Only the X(U) axis moves need to be entered in subsequent blocks after the original G92 command. A G92 command for straight thread cutting is written in the G92 X(U)___ Z(W) ___ F lead format. F is the threading lead feed rate.
- **G96:** Constant surface speed control
- **G99:** Feed per revolution

3.8.2 M Codes (Miscellaneous Functions)

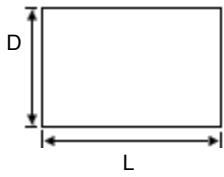
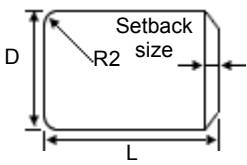
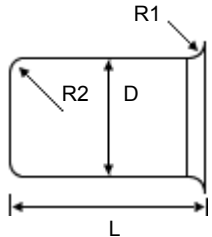
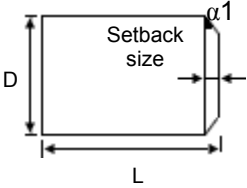
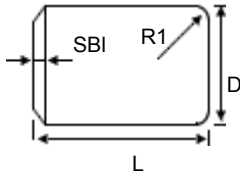
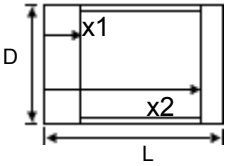
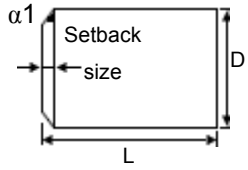
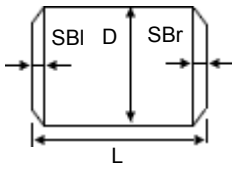
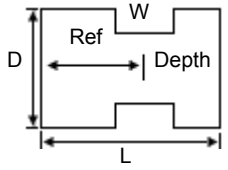
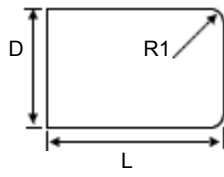
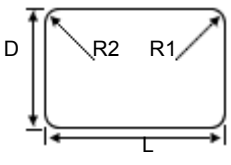
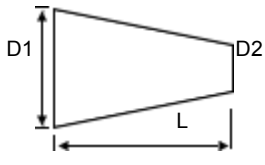
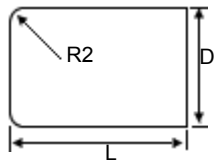
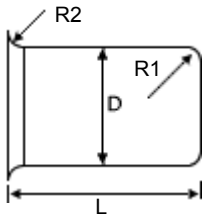
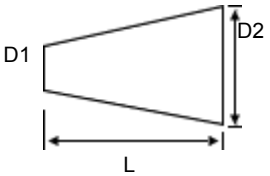
- **M03:** Spindle forward (clockwise)
- **M04:** Spindle forward (counter clockwise)
- **M06:** Automatic tool change
- **M10:** Chuck open
- **M11:** Chuck close
- **M38:** Door open
- **M39:** Door close

3.9 Cutting Tools

Cutting tools are automatically selected for different cutting operations. Used cutting tools and corresponding cutting tool index numbers are shown in Table 3.3. If the number of drill operations are more than three than the new drill tools are replaced with the current drill positions starting from the Drill#1.

Table 3.3 Cutting tools and tool index numbers

Cutting Tool Name	Tool Index No
Right hand cutting tool	T1
Center Drill	T2
Grooving and parting tool	T3
Drill#1	T4
Threading tool	T5
Drill#2	T6
Left hand cutting tool	T7
Drill#3	T8

<p>Straight Cylinder</p> 	<p>Right-End Chamfer + Left-End Convex Fillet</p> 	<p>Right-End Concave Fillet + Left-End Convex Fillet</p> 
<p>Right-End Chamfer</p> 	<p>Right-End Convex Fillet + Left-End Chamfer</p> 	<p>Thread</p> 
<p>Left-End Chamfer</p> 	<p>Both End Chamfer</p> 	<p>Rectangular Groove</p> 
<p>Right-End Convex Fillet</p> 	<p>Both End Convex Fillet</p> 	<p>Converging Taper</p> 
<p>Left-End Convex Fillet</p> 	<p>Right End Convex Fillet + Left End Concave Fillet</p> 	<p>Diverging Taper</p> 

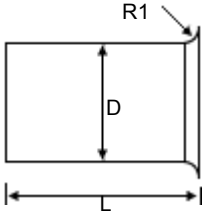
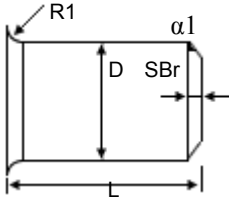
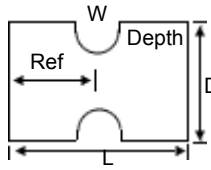
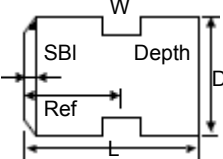
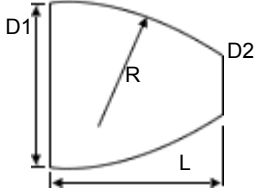
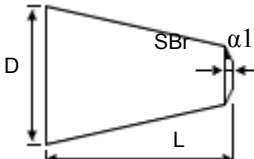
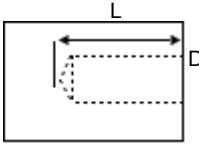
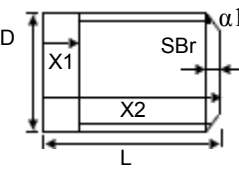
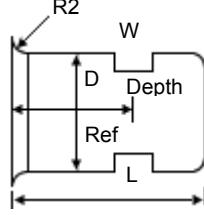
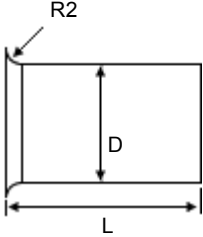
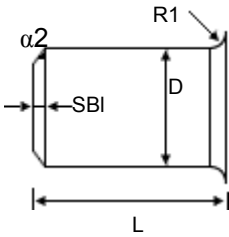
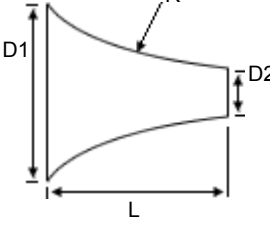
<p>Right-End Concave Fillet</p> 	<p>Right End Chamfer + Left End Concave Fillet</p> 	<p>Round Groove</p> 
<p>Left Chamfer+ Rectangular Groove</p> 	<p>Convex</p> 	<p>Converging Taper+ Right Chamfer</p> 
<p>Drill</p> 	<p>Right End Chamfer +Thread</p> 	<p>Left End Concave Fillet + Rectangular Groove</p> 
<p>Left-End Concave Fillet</p> 	<p>Right-End Concave Fillet + Left-End Chamfer</p> 	<p>Concave Feature</p> 

Figure 3.11 Design features for rotational parts

CHAPTER 4

APPLICATION DEVELOPMENT

Microsoft's Distributed interNet Applications (DNA) architecture is applied to system. A three-tiered framework is build for implementing a web and client/server integrated application.

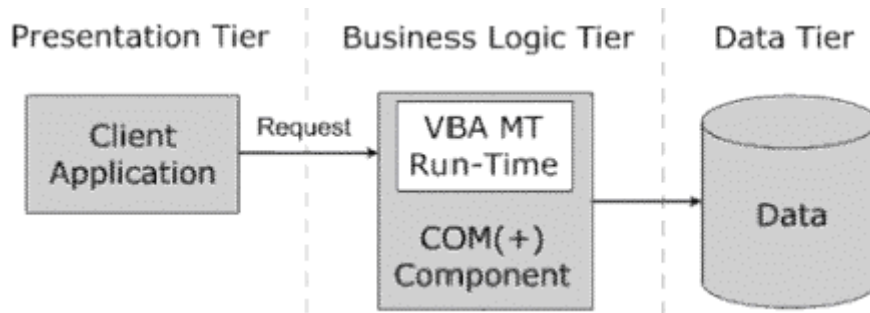


Figure 4.1 DNA Architecture

4.1 Presentation Tier

Presentation tier is the part of the system that the user sees. HTML pages and ActiveX control for 2D wire frame part sketching are the built elements in this tier. Visual Basic 6 is used in the control development. This tier runs on the client side computer. Figure 4.2 demonstrates the web site diagram of the application.

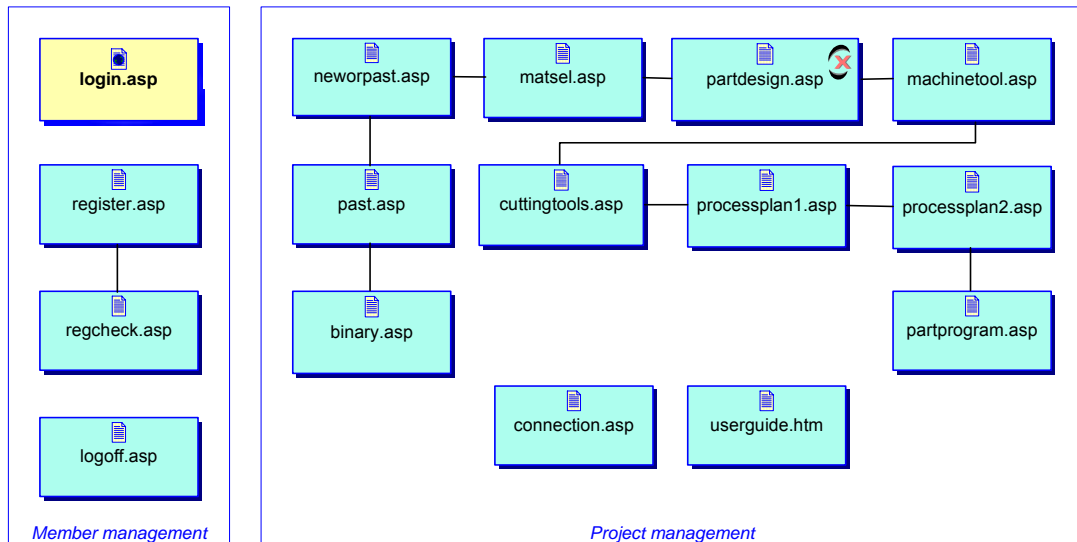


Figure 4.2 Web site diagram of the application

4.2 Business Logic Tier

This tier is composed of services and components that run on the server to implement the specific logic required for a business solution. With Windows 2000, COM+ that is the extension of COM, forms the building structure of business tier. Interpretation of the part data file from database, process plan and part program generation tasks are handled by the created COM+ component. COM dll files are also coded with Visual Basic 6. The WebTURN includes a membership system. This membership system is coded with ASP scripting. The registration of the personal details to the database, authorizing the user name and password in login process, bringing the materials and cutting parameters list to the user's in front and refreshing the cutting parameters with different choice of materials, automatic file name assignment to the product data file, automatic creation of the class instances for business processing are all done by ASP scripting.

4.3 Data Tier

The data tier includes the essential data sources of the system. These sources can be relational database systems. Microsoft SQL Server 2000 is chosen for

database and the system database is built on it. The data tier is composed of six tables:

- **Members Table:** All registered user details which are username, password, name, surname, title, address, phone and e-mail, recorded to this table. Whenever there is a need to reach any information regarding to a certain member the related data is retrieved from this table. For example, the values of username and password for authorizing the user to enter the site are retrieved from ‘members’ table.
- **Materials Table:** Some common materials and corresponding proper cutting parameters are entered to the database. All the content of the materials table is displayed to the user on material selection page of the web site.

The materials and their cutting parameters already recorded into the table are the followings:

Table 4.1 Materials and chosen cutting parameters

Material	Feed Rate (mm/rev)		Depth of Cut (mm)		Recommended Cutting Speed (m/min)		Specific Cutting Energy (GJ/m ³)
	Rough	Finish	Rough	Finish	Rough	Finish	
Alloy Steel	0.25	0.1	2	0.6	50	60	3.5
Aluminum Alloy	0.4	0.1	3	1	50	60	0.6
Brass	0.2	0.1	2	0.5	50	60	0.7
Cast Iron	0.25	0.1	2	0.6	50	60	1.5
Copper Alloy	0.4	0.1	3	1	50	60	1
Mild Steel	0.25	0.1	2	0.6	50	60	2

- **Blank Material Table:** The available blank diameters for the related materials are kept in this table.
- **Parts Table:** The part data file submitted by the user is kept in this table in binary format. The name of the part file is automatically assigned and also written to FileName field of the table.

- **Process Plan Table:** The machining instructions for the corresponding parts are written into this table.

- **Part Program Table:** NC program required to manufacture the part on the Denford CNC Turning Machine is kept in Part Program table.

Database diagram is shown in Figure 4.3.

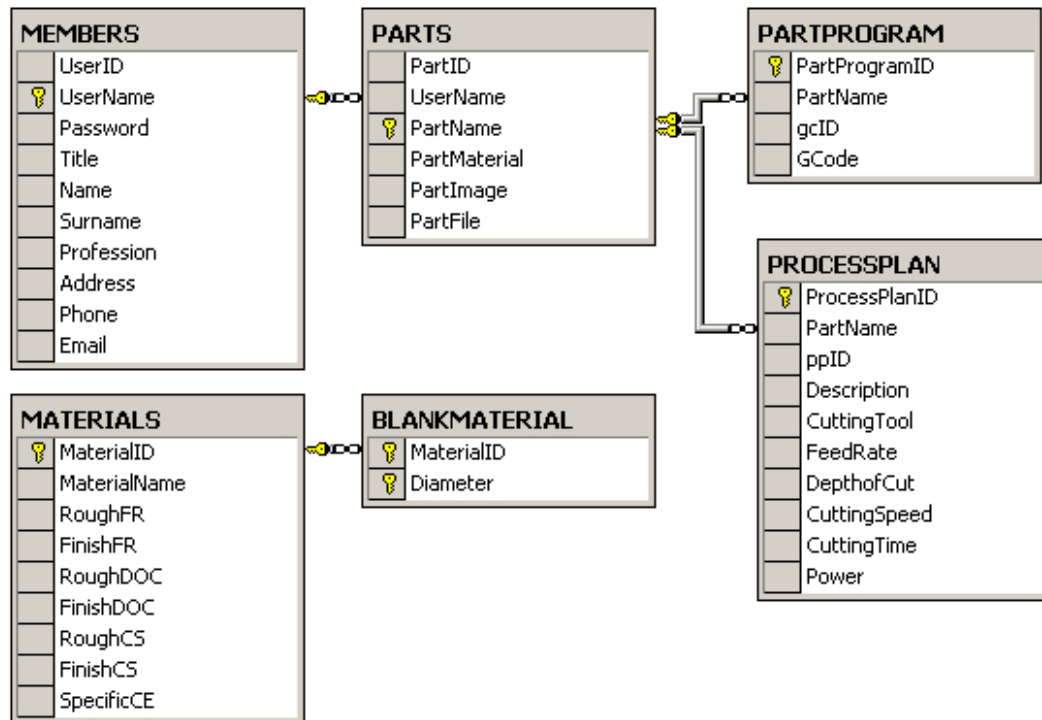


Figure 4.3 Database diagram of WebTURN

The system's entire DNA model is illustrated in Figure 4.4 and the UML class diagram of the business logic is depicted in Figure 4.5.

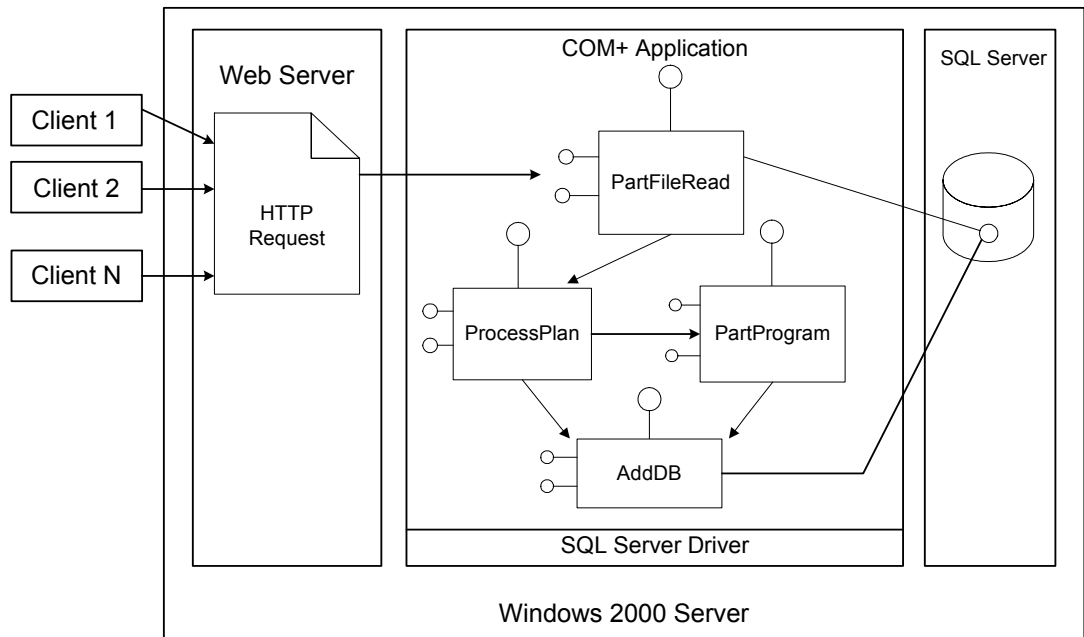


Figure 4.4 DNA Model of WebTURN

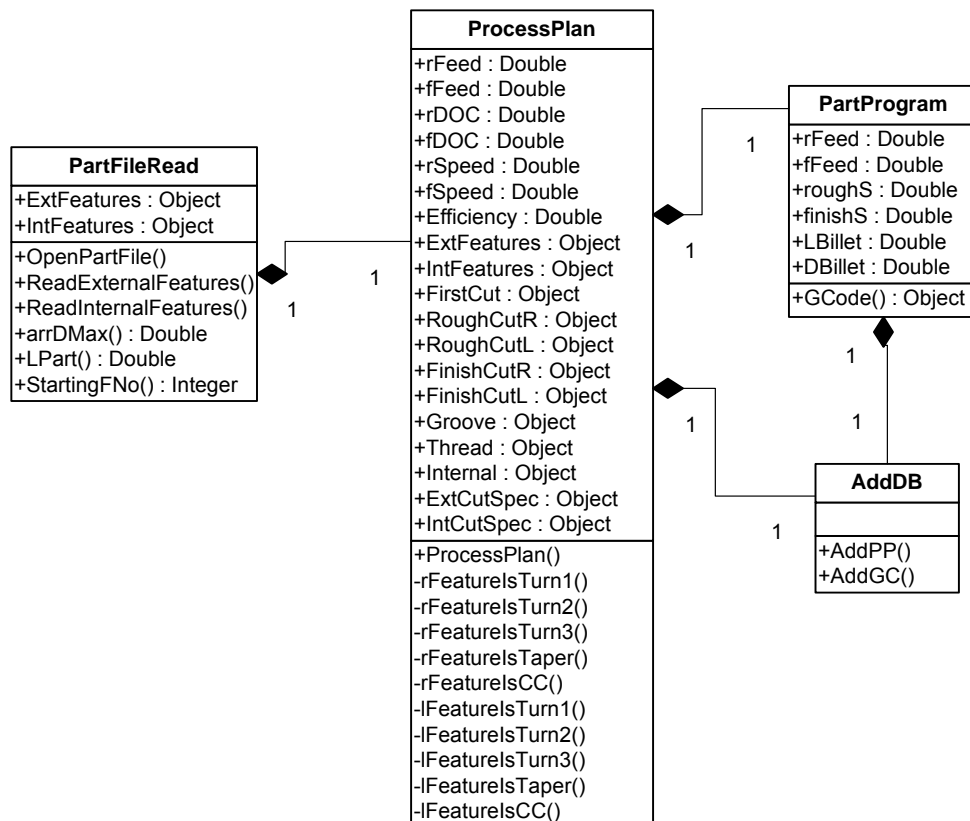


Figure 4.5 UML class diagram of the business logic

4.4 Part Design Control Development

The user interface control development for the geometric part design was the one of the most challenging and crucial stages in this work. There is a need to make a graphical user interface to enter a number of design specifications for a number of design features in a compact area. Besides that, the real time appearances of the features should be drawn on to the sketching window. That gives a better visualization facility to understand whether the inserted feature is correct. If not, the feature insertion can be taken back in the early design phase unlike to AutoTURN. For this purpose an ActiveX control is developed in the Visual Basic environment. The lack of built in graphical capabilities of the Visual Basic enforce the use of a third party controller. Thus, a ready-made controller named Flipper CAD controller is utilized to overcome the difficulty in computer graphics development. It has its own set of attributes, methods and events, which are invoked by coding. Part sketching window is constructed with this controller. It is possible to display 2D wire frame geometries on sketching window. On developed user interface, a toolbar is situated at most left hand site for primary design features. Users can select the features by clicking on the icon of the features on the toolbar then the data entry window for the related feature is opened right below the sketching window. The user interface of the sketching window together with the data entrance window for turn feature is depicted in Figure 4.6.

The part length and the diameter are the compulsory dimensions in defining a turn feature. Tolerance entry is also possible for both major dimensions.

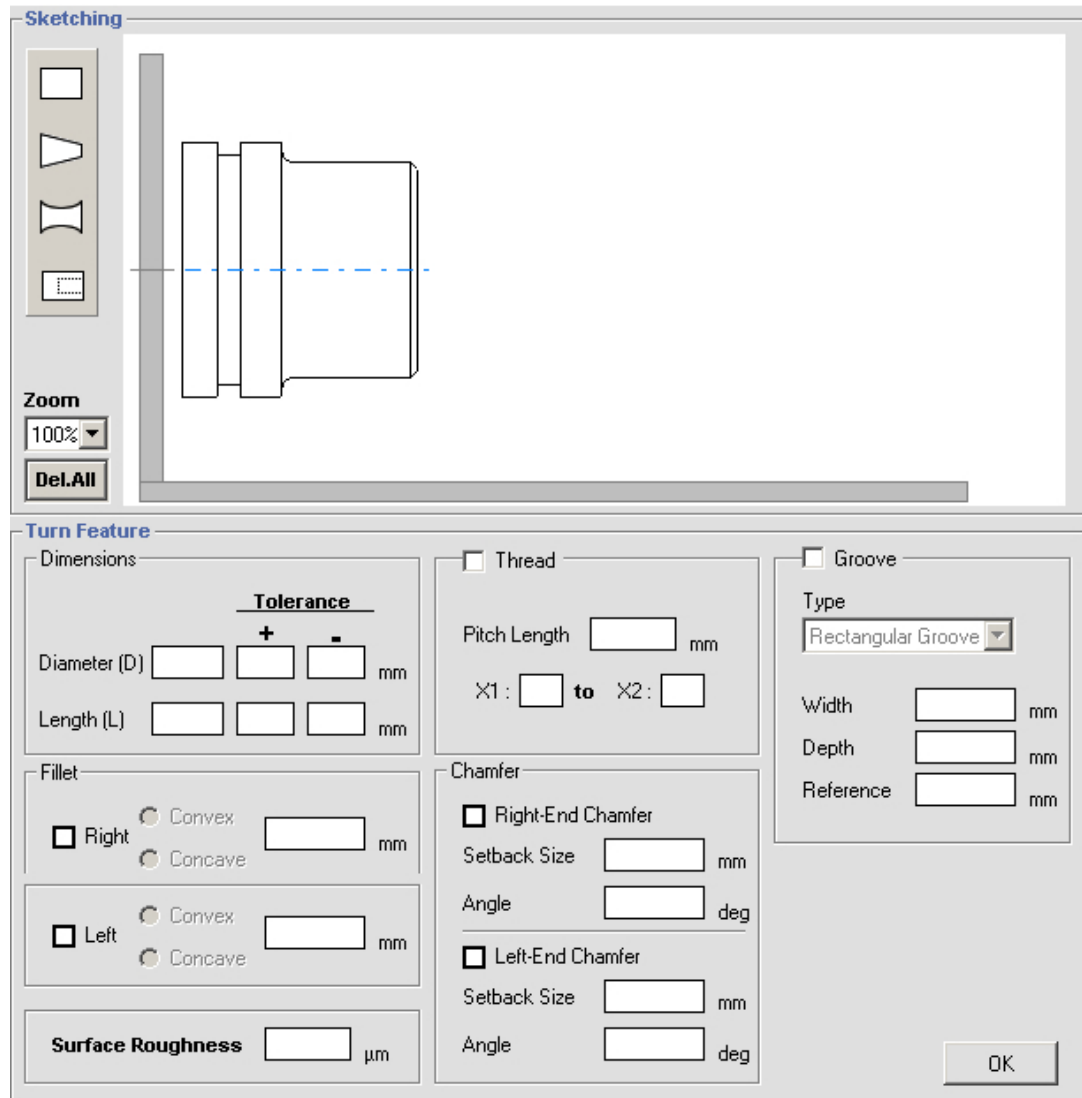


Figure 4.6 ActiveX component and data entrance window

The tolerance values are used to check whether or not the diameter and length values satisfy the design specifications after manufacturing. Denford CNC turning machine has the limitation to turn workpieces of maximum length of 180 mm and diameter of 100 mm. While the features are attached sequentially, the required billet diameter and length are calculated and checked automatically whether it is less than, equal to or greater than this limit. If an entered value does not satisfy the limitations a warning message appears requesting user to change it.

Fillet is one of the sub-features that can be defined on the turn features. Right end convex or concave and left end convex or concave features can be added.

Radius is required for fillet definition. The corresponding radius entrance text boxes become active when right or left fillets are selected. When fillets are not selected, it is not possible to enter a radius by mistake.

Chamfer is another sub-feature that can be defined on the turn feature. The setback size and the chamfer angle are the required parameters for chamfer definition. The interface enables the users to add both right and left end chamfers. Similarly, the data entry text boxes for chamfer parameters are inactive unless the user selects one of the chamfers. The controller prevents user from inserting both chamfer and fillet sub-features to the same end. In case of such a conflict, user is warned with a message.

One another sub-feature is the thread. The necessary parameters to define a thread are the pitch length and the thread starting and finishing points. Those points are referenced to the starting point of the feature that is most left end of the feature.

Grooves might also be added as a sub-feature. Two types of grooves, rectangular and round grooves can be defined on a turn feature. To define a groove, depth and width of the groove and groove reference are the needed parameters. Herein, the groove reference is the distance of the mid-point of the groove to the starting point of the feature. The parameter entry text boxes for grooves stay inactive until groove is selected from the groove checkbox.

Additionally, user may enter a surface roughness value for the feature. Surface roughness does not have an effect on manufacturing in this work.

The second primary feature is the taper feature. The sketching window and data entry window of the taper features are shown in Figure 4.7. Two types of taper features are definable in sketching the taper features, which are converging and diverging types. The required basic parameters to define a taper feature are the starting (D1) and finishing (D2) diameters and the length (L) of the feature. Tolerances for these basic parameters can be defined through the user interface.

Right and left chamfers can be added to the converging and diverging features. Surface roughness may also be stated for the taper features.

The third and fourth primary features are the convex and concave features respectively. Figure 4.9 shows a view of convex and concave features. Starting diameter (D1), ending diameter (D2), feature length (L) and the radius (R) of convexity or concavity are the required parameters for the definition of convex or concave features.

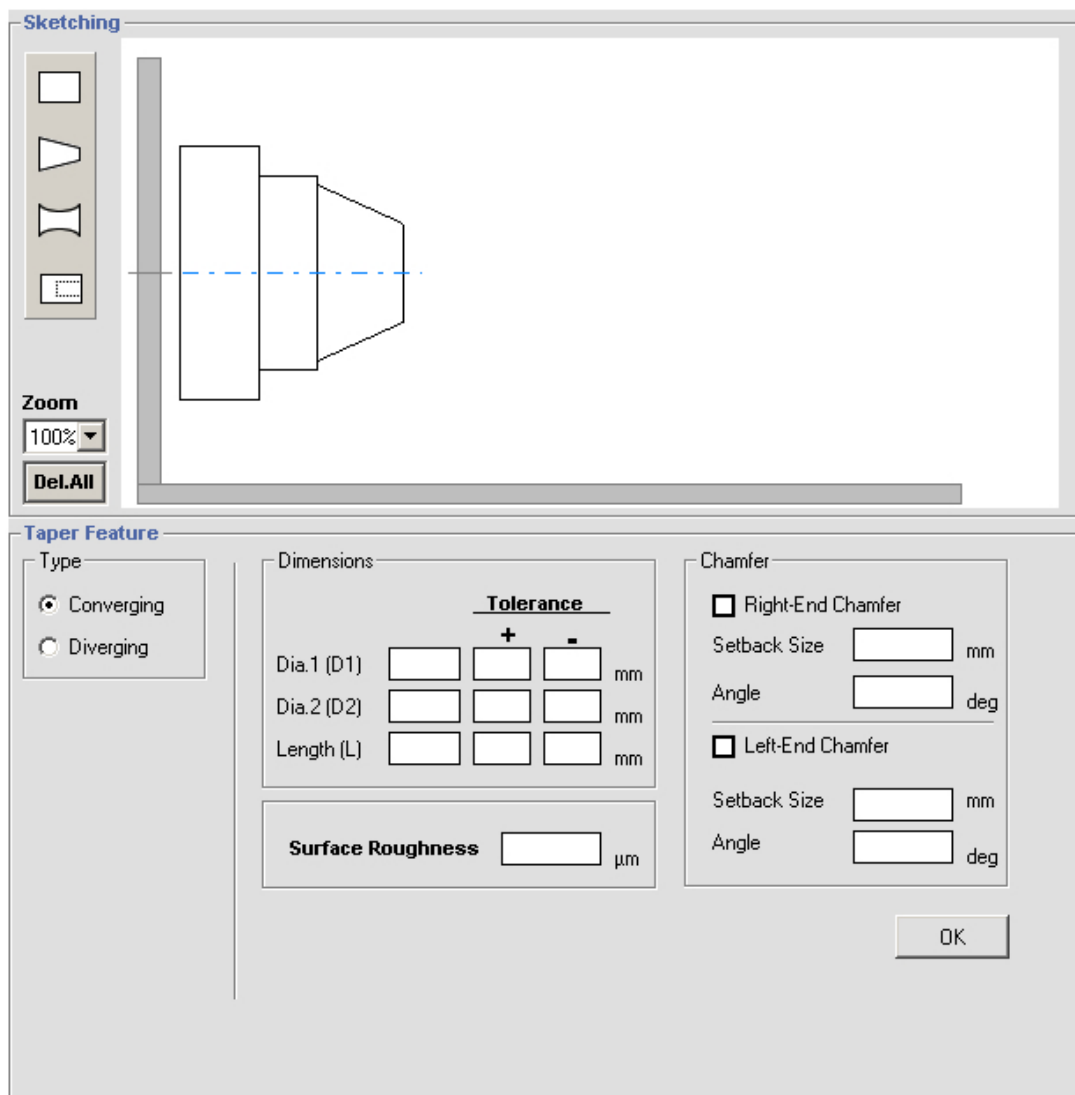


Figure 4.7 Sketching and taper feature data entrance windows

The sketching of convex and concave features requires more attention than other features. Since the center coordinates of the arcs are not known in advance, these two points should be calculated by the controller to draw the feature correctly.

Two quadratic equations come into existence and needs to be solved by the computer program. The center points might be in one of the three different positions, at the left of the feature, between the major and minor diameters and at the right of the feature. Those three possibilities should be taken into consideration for both features.

The equations to be solved by the computer for each convex or concave feature insertion are:

$$x^2 + y^2 = r^2 \quad (4)$$

$$(L-x)^2 + \left(y + \frac{(D1-D2)}{2}\right)^2 = r^2 \quad (5)$$

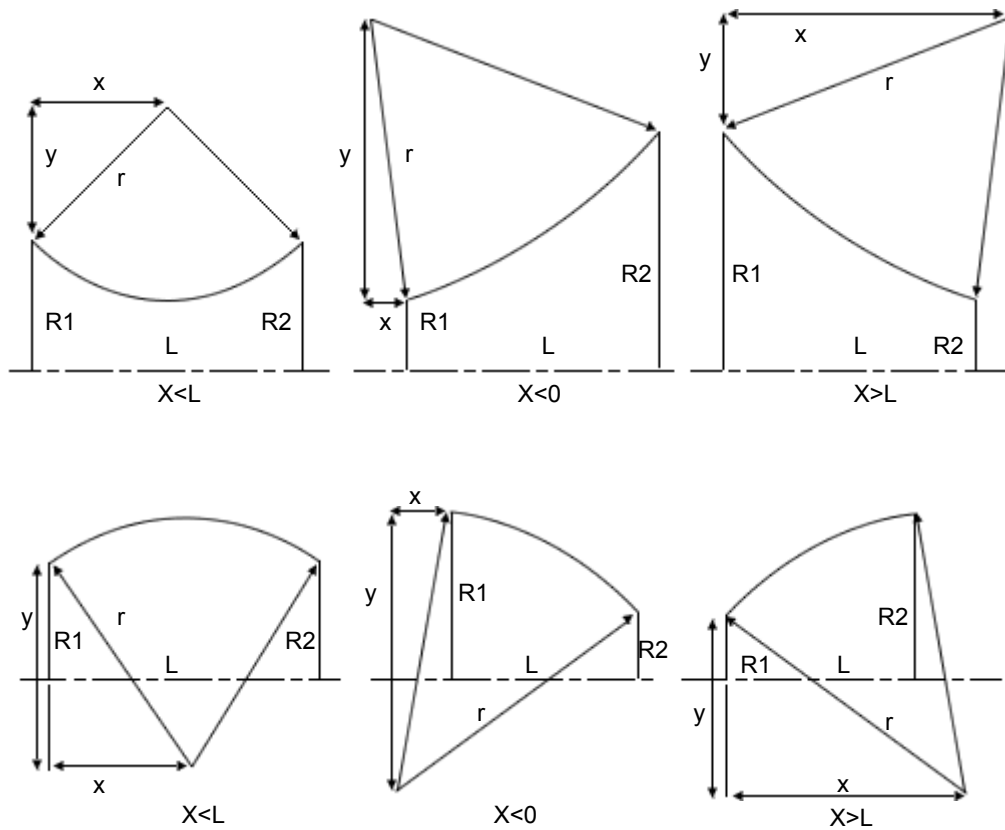


Figure 4.8 Conditions of the convex & concave features

When the user control is installed on to the user's computer. The part file name is automatically assigned. If a registered user has already defined five features prior to the current part design, then the present design is named as "username6". Here, the user's username and the sequencing part number are concatenated. When the whole geometric design is completed, user should click on the part complete button at the right most bottom of the controller. After clicking, a text file naming 'username6' is created at the application path of the user control in the client computer. Then this file is saved into the database 'Parts' table. The only way to record files into databases is converting the file's ASCII (American Standard Code for Information Exchange) format to binary data format.

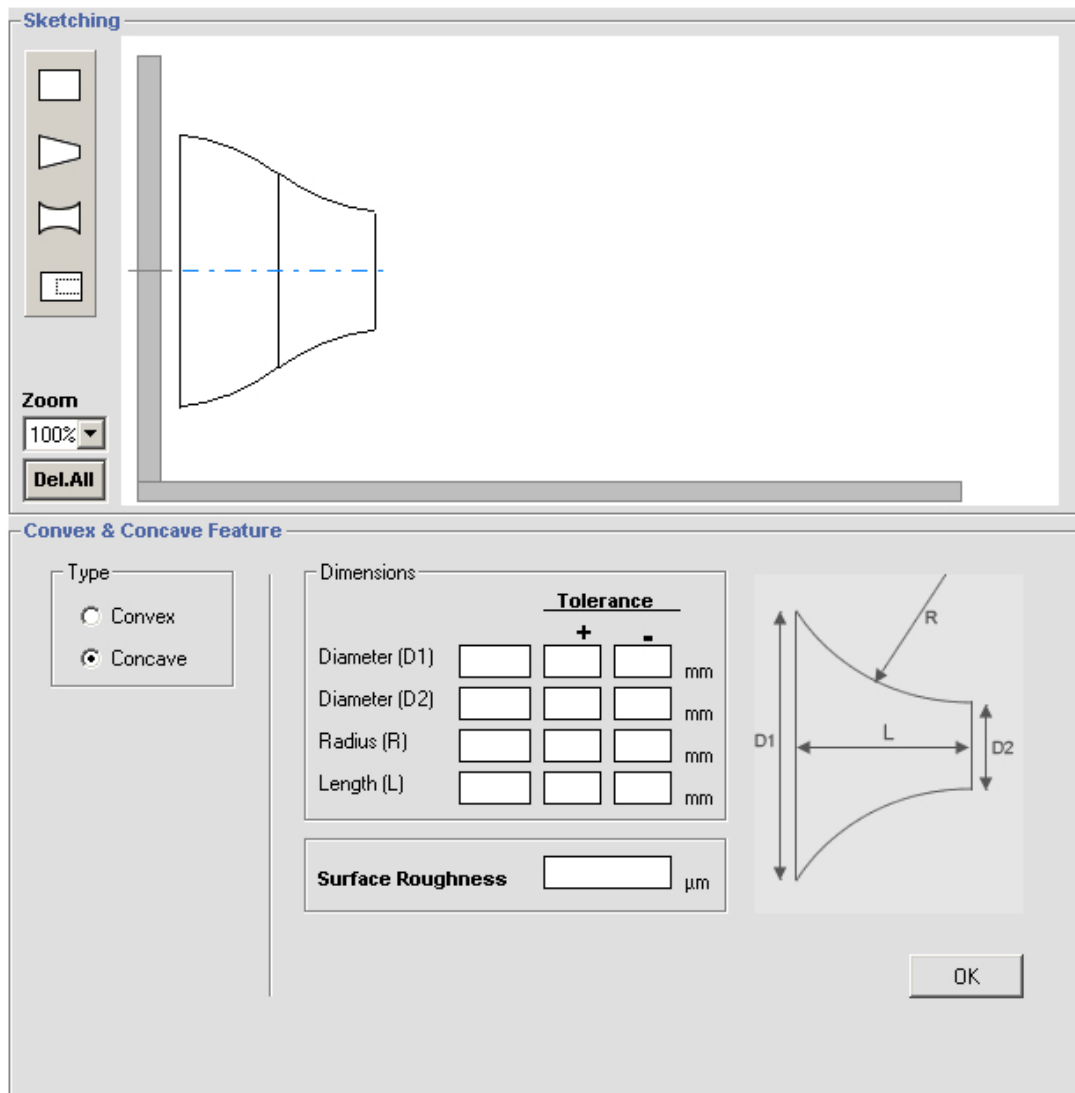


Figure 4.9 Data entrance window for convex and concave features

Drill features can be defined as internal features after external feature insertion is completed. Drill diameter, length and drill bottom condition are the required parameters to define a drill. Drill definition window is shown in Figure 4.10.

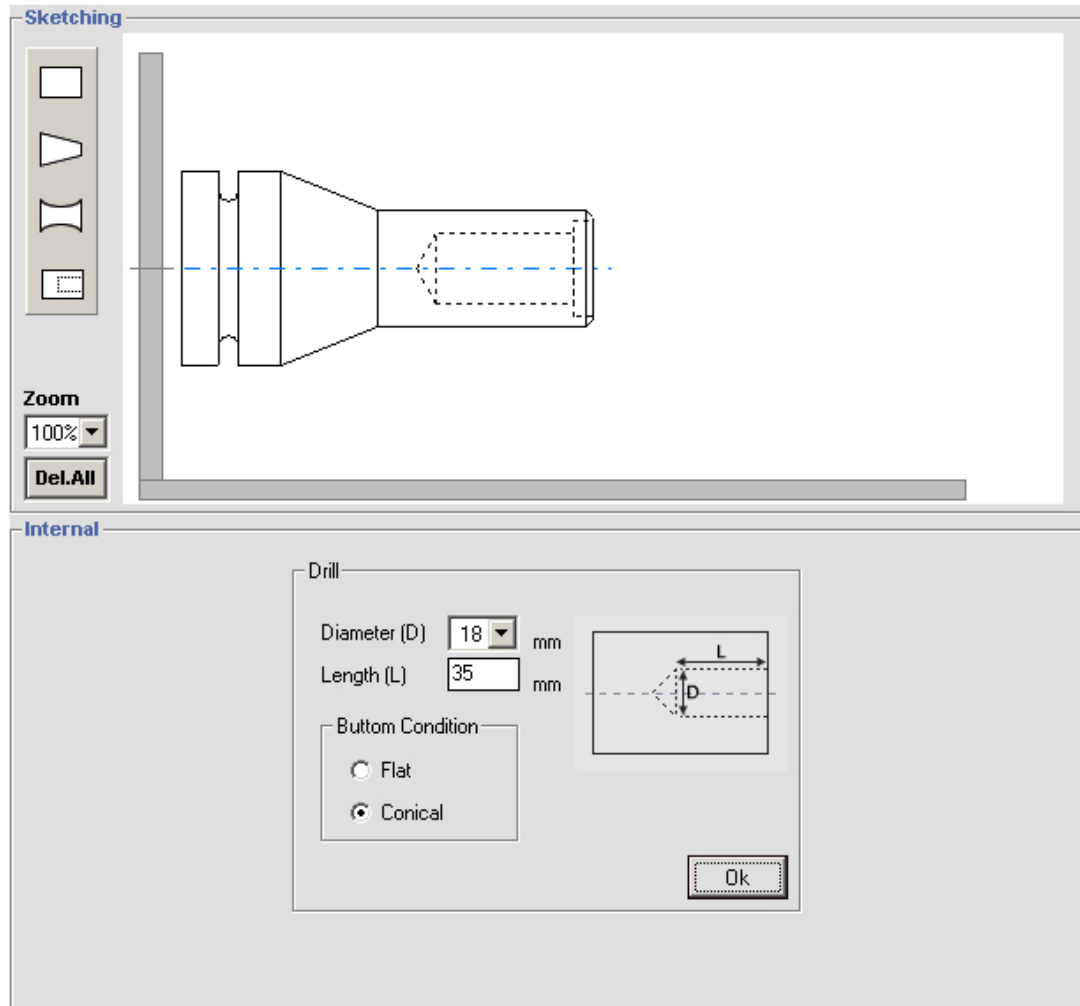


Figure 4.10 Data entrance window for drill feature

4.5 Reading of Product Data File

After the part drawing file is recorded into the database, customized COM+ server component for part file reading is initiated. At the same time the information of submitted part file name is passed through the component. Then, this component connects to the “Parts table” in the database and retrieves the corresponding part file. Firstly, it converts the binary data format of the file to the ASCII text format

then it starts reading. Feature recognition is accompanied with part reading. The knowledge of each feature recognized is added to external or internal features dictionary objects that were created at the beginning of the process.

4.6 Process Planning

Process planning class is instantiated right after the part file reading is completed and associated class is destroyed. Process planning class has public attributes, which are roughing and finishing feed rates, depth of cuts, cutting speed, machine tool motor efficiency, specific cutting energy of the workpiece material. Appropriate values are assigned to those attributes just before the process planning procedure. Previously, part read class outputs, which are the list of features, part length and largest diameter become the parameters of process planning sub routine. At the beginning of process planning, coordinates of the nodal points for all features are determined according to the coordinate system of the CNC lathe.

4.7 Part Programming

When the process planning class completed its job, it is destroyed and part programming class is instantiated. The output dictionary objects of the process planning which carry the knowledge of cutting processes and cutting paths become the input parameter of the G Code subroutine of part programming class. When the G Code subroutine is completed, it returns back the collection of machining instructions for the CNC lathe.

CHAPTER 5

TEST RUNS

This chapter exemplifies three complete case studies. It shows the steps from early user registration to the generation of part program.

5.1 Sample Run 1

For example, a visitor to the WebTURN's website aiming to manufacture a rotational part, is supposed be an unregistered user. From the link provided in the login home page, user is directed to registration page. The visitor is than expected to fill the registration form like in the Figure 5.1.


When the registration is completed, user goes to the login page and enters the username and password. After authorization, the user is directed to the new or past project selection page. If the user clicks on the new project button he is directed to the part name entry page. It is required to enter a unique part name in this page. After that, he is directed to material selection page. User can then select a desired material from the available list. The log in, new or past project selection, part name entry and material selection pages are shown in Figures 5.2, 5.3, 5.4 and 5.5.

WebTURN : Registration - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print View Source

Address http://cim.me.metu.edu.tr/webturn/register.asp



Project
WebTURN

December 18, 2003

Log in
Register

User Guide

Please fill the following blanks.

Title: Mr
Name: Erhan
Surname: Özşüer
*Username: erhan
*Password: *****
*Reenter Password: *****
Profession: Mechanical Engineer
Address: METU Industrial Engineering Department, No:302
Phone: (312) 210 46 93
*E-mail: erhan@ie.metu.edu.tr

Register

*Fields must be entered!


Figure 5.1 Run-1: Registration page

WebTURN : Log in - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print View Source

Address http://cim.me.metu.edu.tr/webturn/login.asp



Project
WebTURN

December 18, 2003

Log in
Register

User Guide

LOGIN

Welcome to project WebTURN website.

WebTURN is an open design and manufacturing environment for wide range of rotational parts.

If you are a registered user, continue with the Log in, if not first [register](#).

Registered User

Username: erhan
Password: *****

Log in

Figure 5.2 Run-1: Log in page

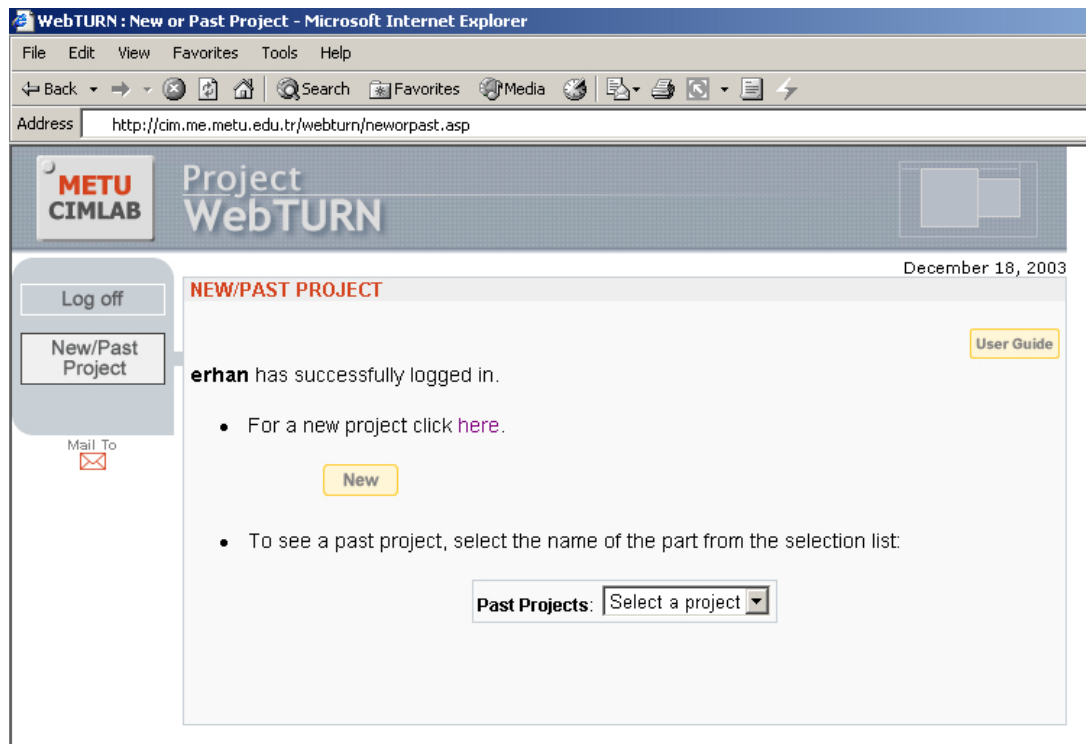


Figure 5.3 Run-1: New or past project selection page

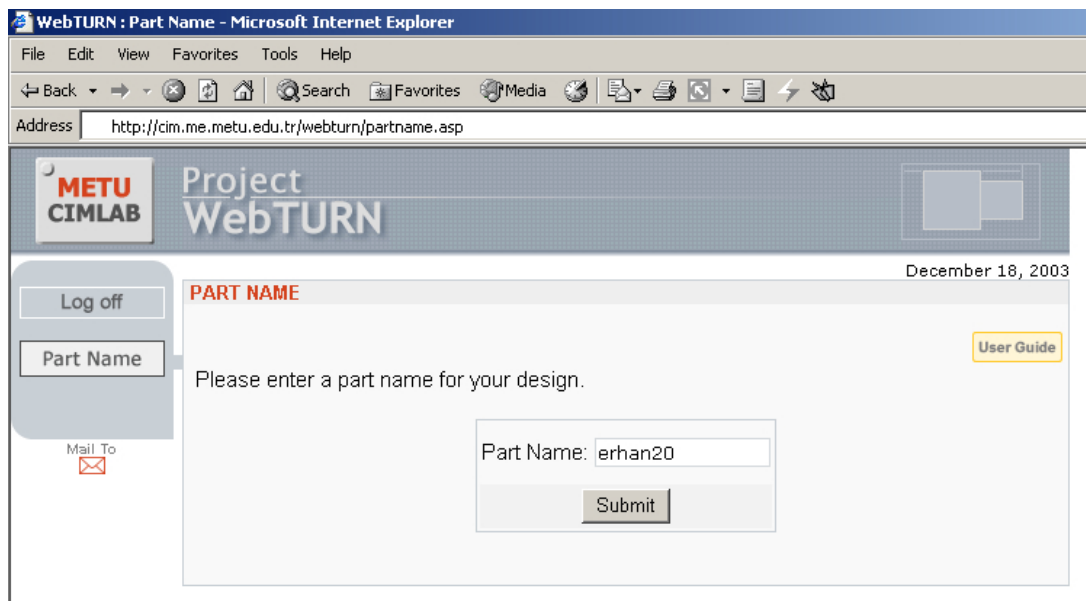


Figure 5.4 Run-1: Part name entry page

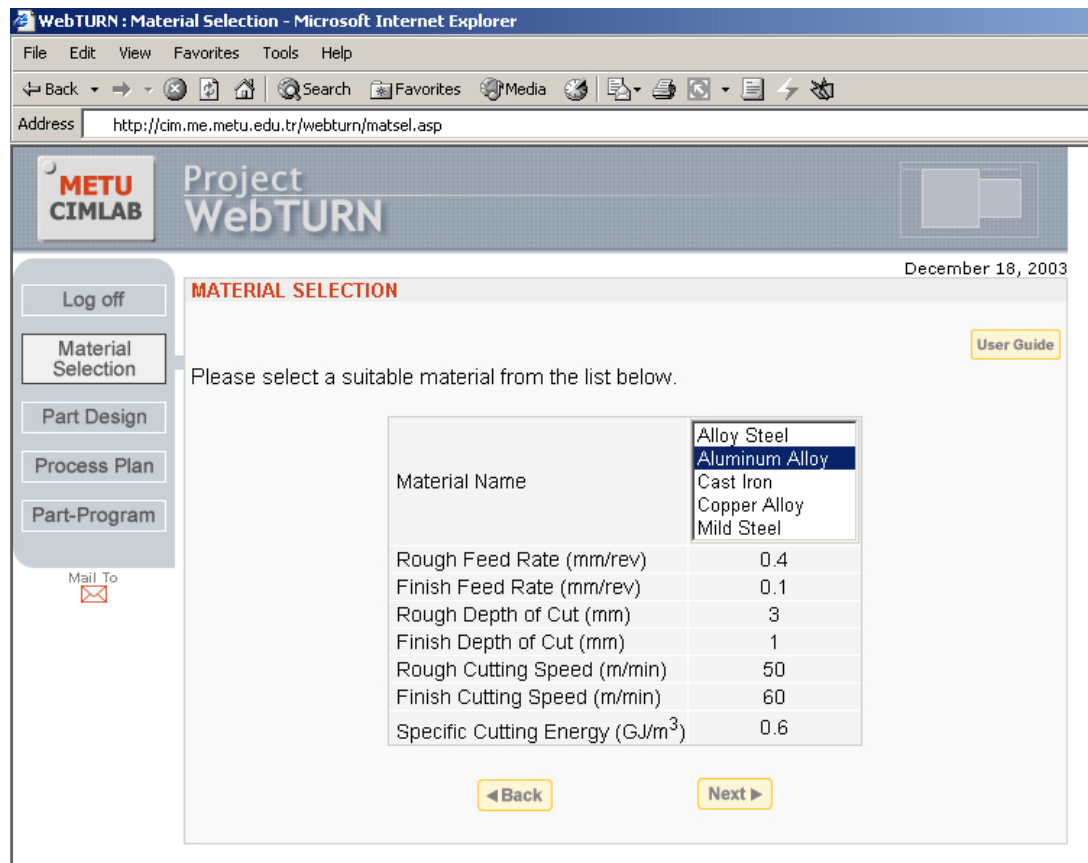


Figure 5.5 Run-1: Material selection page

Let us assume that the chosen material is aluminum alloy and the user continues with the part design. The engineering drawing of the part that is going to be sketched is shown in Figure 5.6. The sketched part in the design stage is shown in Figure 5.7 below.

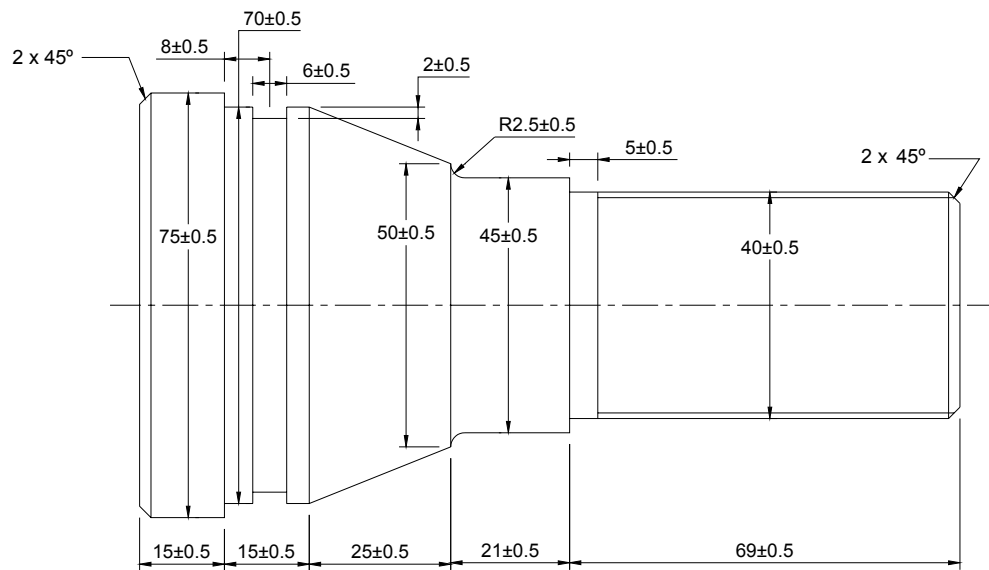


Figure 5.6 Run-1: Engineering drawing of the part (units in mm)

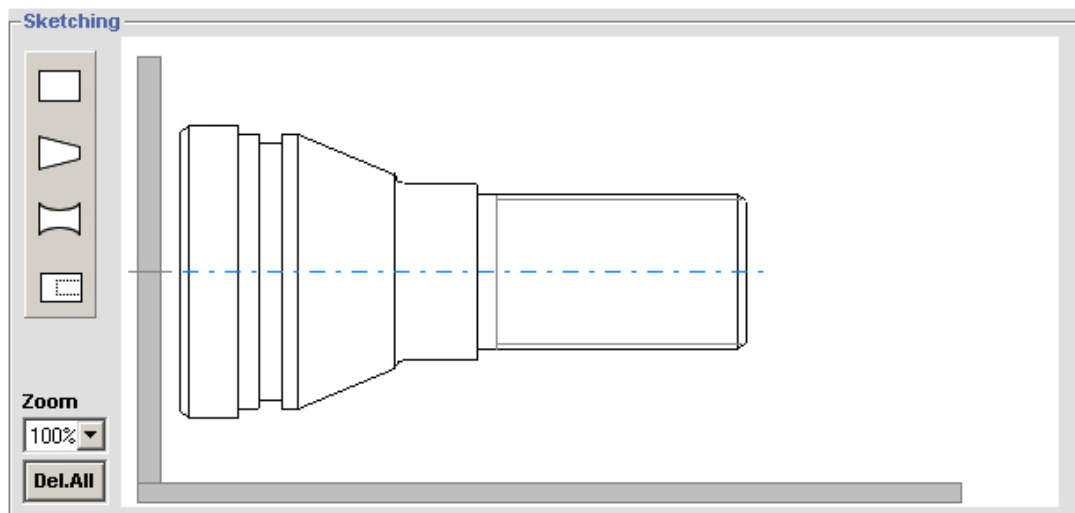


Figure 5.7 Run-1: Sketched part in the design stage

After the user sketched the part, the part data file is created in the user's computer and then saved to the database. The ASCII format of the file before saving into the database is shown in Table 5.1.

Table 5.1 Run-1: Part data file

```
"Owner","Erhan Özsüer"
"Creation Date","12/18/2003"
"Feature","Turn"
"Type","LChamfer",0
"D",75,.5,.5
"L",15,.5,.5
"LChamfer",2,45
"Feature","Turn"
"Type","RectangularGroove",0
"D",70,.5,.5
"L",15,.5,.5
"RectGroove",6,2,8
"Feature","Taper"
"Type","Converging_Taper",0
"D1",70,.5,.5
"D2",50,.5,.5
"L",25,.5,.5
"Feature","Turn"
"Type","LHConcaveFillet",0
"D",45,.5,.5
"L",21,.5,.5
"LHConcaveFillet",2.5
"Feature","Turn"
"Type","RChamfer_Thread",0
"D",40,.5,.5
"L",69,.5,.5
"RChamfer",2,45
"Thread",5,69,1.5
```

After the part data file is received from user, total part length is calculated and the maximum part diameter is found out by interpreting the part data file. All possible blank diameters are retrieved from the “BlankMaterials” table and the shortest possible blank length is calculated by considering the portion in the chuck, and facing, cutting off and safe margin lengths. And then those values are returned to the “billet definition and cutting parameters edit page”. The user can alter the blank diameter and length values in this page. Cutting parameters concerning to the chosen material are also displayed in this page, user can modify those values. The billet definition and cutting parameters edit page is shown in Figure 5.8.

Project WebTURN

December 18, 2003

Log off

Material Selection

Part Design

Process Plan

Part-Program

PROCESS PLAN

User Guide

Designed Part Specification

Part Name:	erhan20
Material:	Aluminum Alloy
Maximum Diameter:	75 mm
Total Part Length:	145 mm

Blank Size

Blank Diameter:	80 mm
Blank Length:	176 mm

Cutting Parameters

Rough Feed Rate	0.4	mm/rev
Finish Feed Rate	0.1	mm/rev
Rough Depth of Cut	2	mm
Finish Depth of Cut	1	mm
Rough Cutting Speed	50	m/min
Finish Cutting Speed	60	m/min
Specific Cutting Energy	0.6	GJ/m ³

Next ►

Figure 5.8 Run-1: Billet definition and cutting parameters edit page

After the user decided on the blank size and cutting parameters, he clicks on the next button. Then, he is directed to the process planning page. This page is shown in Figure 5.9. In this page, the used cutting parameters and the estimated machining values for all individual cutting processes are displayed sequentially.

WebTURN : Process Plan - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print View Source Reload

Address http://cim.me.metu.edu.tr/webturn/processplan2.asp

METU CIMLAB Project WebTURN

December 18, 2003

Log off

Material Selection

Part Design

Process Plan

Part-Program

Mail To

PROCESS PLAN

User Guide

Operation	Tool	Feed rate (mm/rev)	Cutting speed (m/min)	DOC (mm)	Time (min)	Power (Watt)
Facing	T1	0.1	60	1	1.68	71
Rough Right Turning	T1	0.4	50	2	4.29	471
Finish Right Turning	T1	0.1	60	1	2.85	71
Grooving, Width=6 mm	T3	0.1	60	2	0.26	141
Threading, Length=64 mm	T5	1.5	50	2	0.85	211
Parting	T3	0.1	60	2	1.68	141

Next ►

Figure 5.9 Run-1: Process plan page

Once the user clicks on the next button in the process plan page, he is directed to the part program page. It is shown in Figure 5.10. In this page the generated part program to machine the part on Denford Mirac Turning Machine is listed in a scrolled text area. All the text shown in this area can be selected and copied to somewhere else. If the user clicks on the new process plan button, he can generate a new process plan and part program for different billet size and cutting parameters. If the user clicks on the new part button, he starts a new project from scratch.

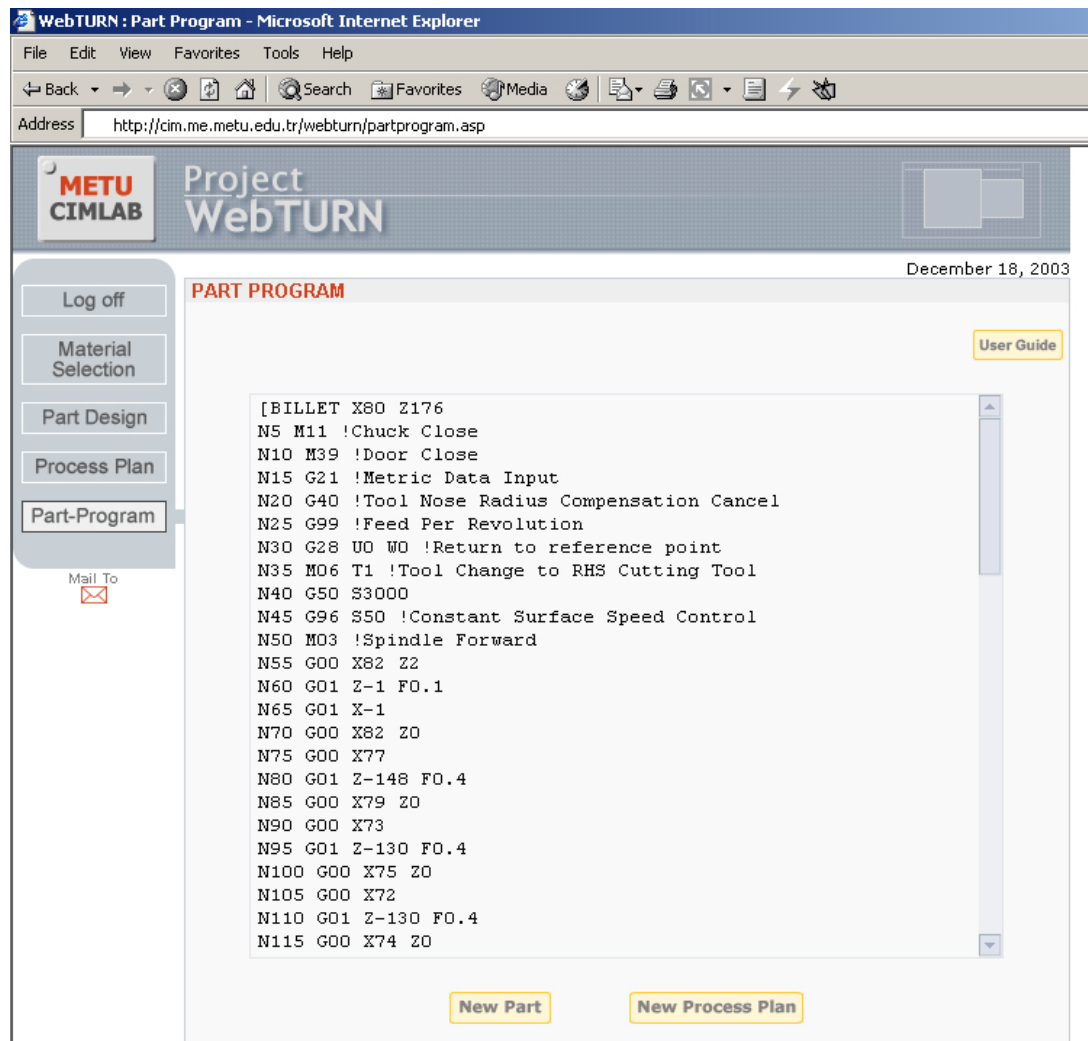


Figure 5.10 Run-1: Part program page

5.2 Sample Run 2 (manufactured)

In this run, the user is assumed to be same with the previous run. After log in, he selects starting a new project. After new project selection, the user is directed to the part name entry page. The part name entry page is shown in Figure 5.11. If the entered part name has already existed in the database, user is warned to change part name. When the part name is validated, user is automatically directed to the material selection page. Material selection page and the chosen material that is brass are shown in Figure 5.12.

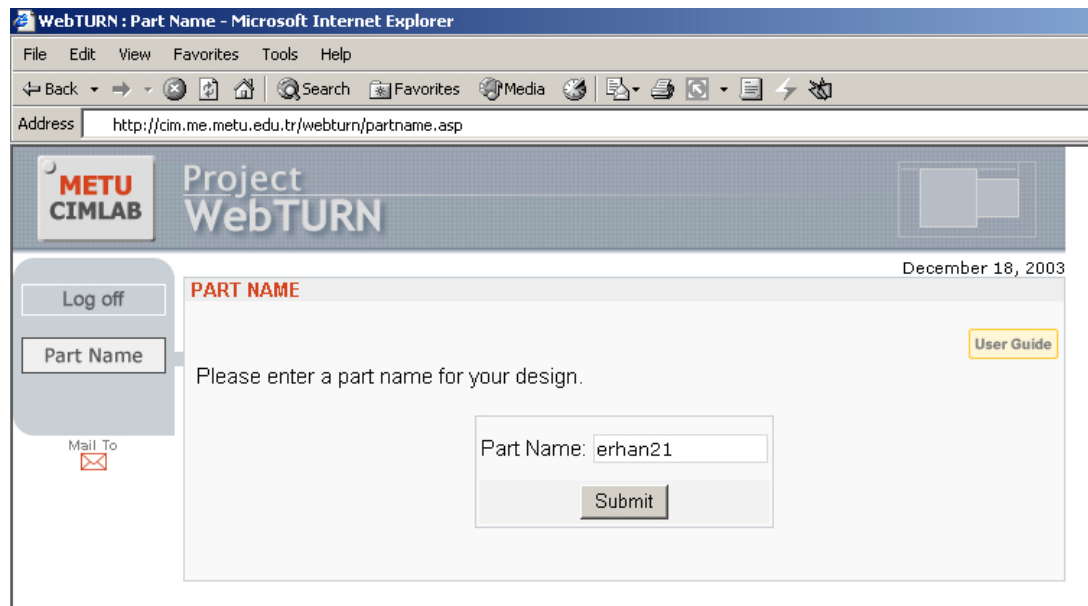


Figure 5.11 Run-2: Part name entry page

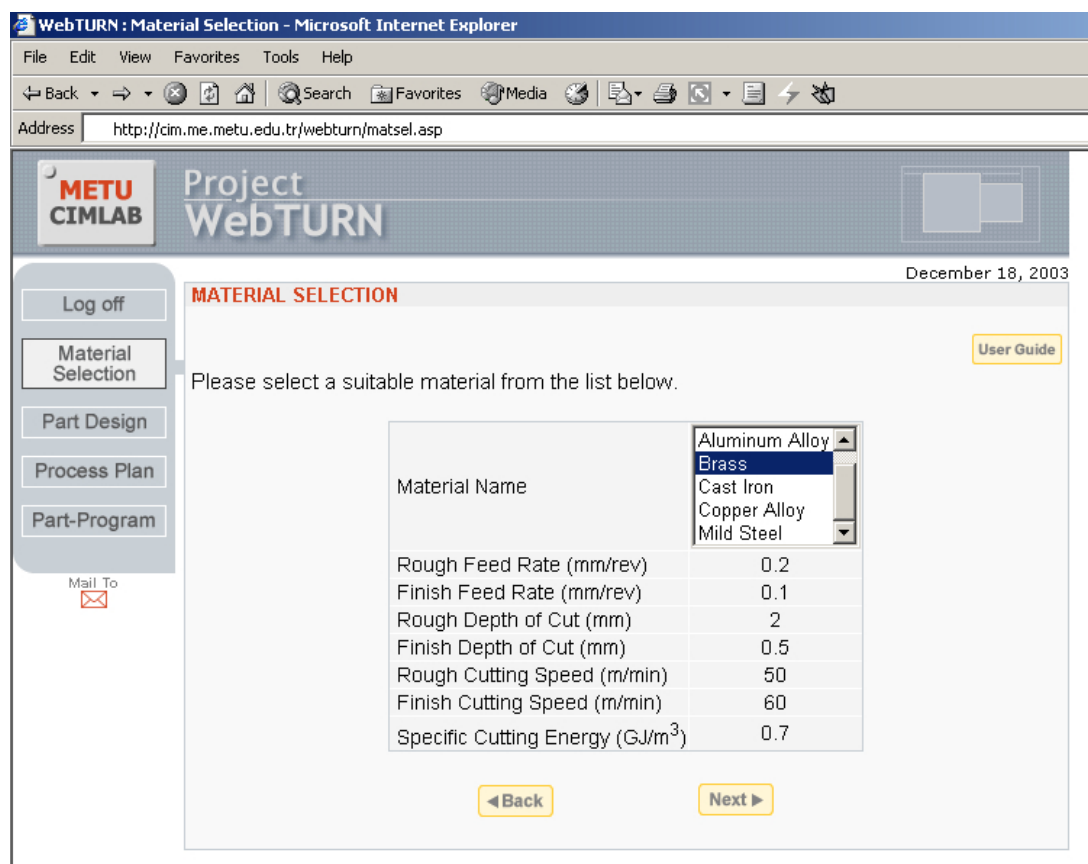


Figure 5.12 Run-2: Material selection page

The engineering drawing of the part that is going to be sketched is shown in Figure 5.13.

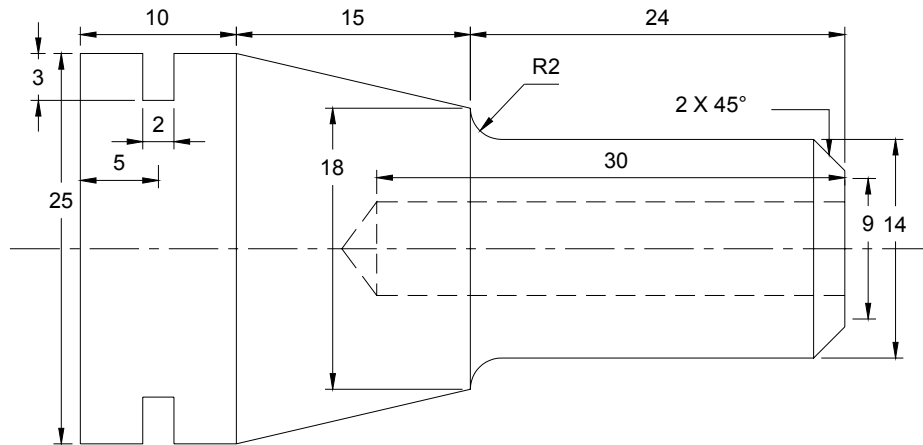


Figure 5.13 Run-2: Engineering drawing of the designed part (units in mm)

Figure 5.14 illustrates the sketched part inside the sketching window.

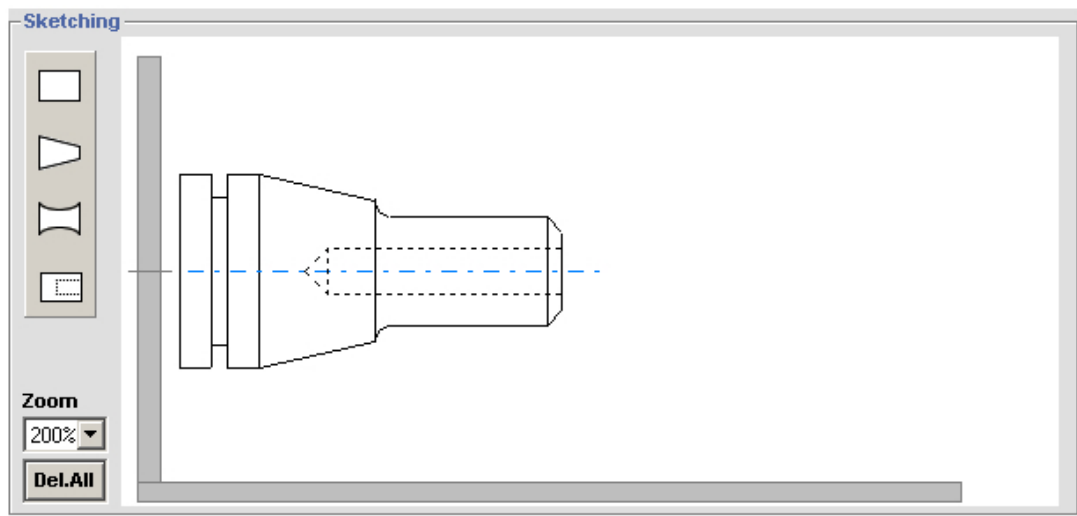


Figure 5.14 Run-2: Sketched part in design stage

Table 5.2 Run-2: Part data file

```
"Owner","Erhan Özsüer"
"Creation Date","12/18/2003"
"Feature","Turn"
"Type","RectangularGroove",0
"D",25,0,0
"L",10,0,0
"RectGroove",2,3,5
"Feature","Taper"
"Type","Converging_Taper",0
"D1",25,0,0
"D2",18,0,0
"L",15,0,0
"Feature","Turn"
"Type","LHConcaveFillet_RChamfer",0
"D",14,0,0
"L",24,0,0
"RChamfer",2,45
"LHConcaveFillet",2
"Feature","Internal"
"Type","ConicalEndDrill"
"D",6
"L",30
```

Table 5.2 shows the ASCII format of the part data file saved to the user's computer.

Figure 5.15 shows the billet definition and cutting parameters edit page.

METU CIMLAB Project WebTURN December 18, 2003

Log off Material Selection Part Design **Process Plan** Part-Program

Mail To

PROCESS PLAN [User Guide](#)

Designed Part Specification

Part Name:	erhan21
Material:	Brass
Maximum Diameter:	25 mm
Total Part Length:	49 mm

Blank Size

Blank Diameter:	30 mm
Blank Length:	80 mm

Cutting Parameters

Rough Feed Rate	0.15	mm/rev
Finish Feed Rate	0.1	mm/rev
Rough Depth of Cut	2	mm
Finish Depth of Cut	0.5	mm
Rough Cutting Speed	150	m/min
Finish Cutting Speed	200	m/min
Specific Cutting Energy	0.7	GJ/m ³

[Next ►](#)

Figure 5.15 Run-2: Billet definition and cutting parameters edit page

Figure 5.16 depicts the process plan of the part.

WebTURN : Process Plan - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address <http://cim.me.metu.edu.tr/webturn/processplan2.asp>

METU CIMLAB Project WebTURN December 18, 2003

Log off Material Selection Part Design **Process Plan** Part-Program

Mail To

PROCESS PLAN [User Guide](#)

Operation	Tool	Feed rate (mm/rev)	Cutting speed (m/min)	DOC (mm)	Time (min)	Power (Watt)
Facing	T1	0.1	200	1	0.07	275
Rough Right Turning	T1	0.15	150	2	0.46	618
Finish Right Turning	T1	0.1	200	0.5	0.2	137
Grooving, Width=2 mm	T3	0.1	200	2	0.01	549
Center Drilling	T2	0.15	150	-	0.01	55
Drilling, D=6	T4	0.15	150	-	0.13	79
Parting	T3	0.1	200	2	0.07	549

[Next >](#)

Figure 5.16 Run-2: Process plan page

Figure 5.17 shows the part program of the designed part.

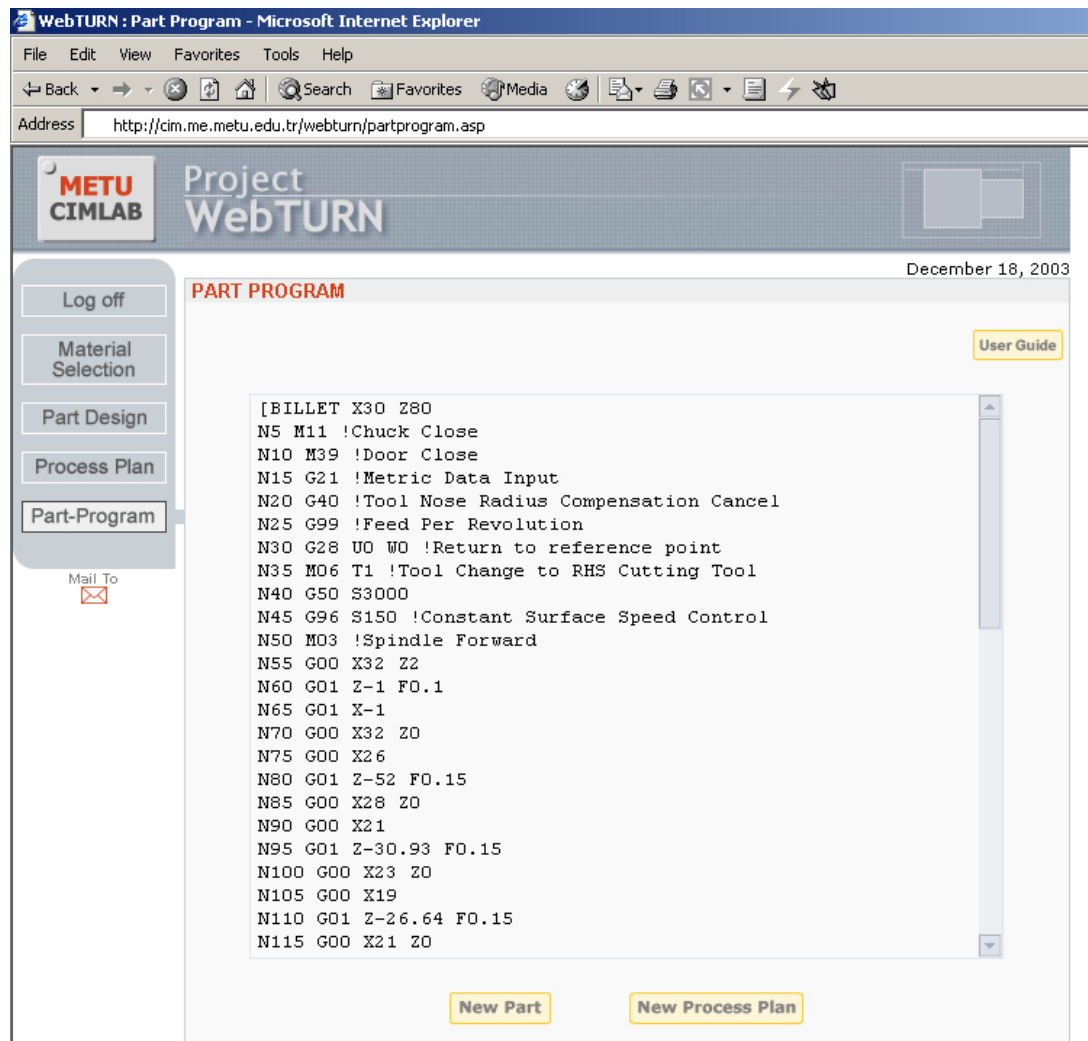


Figure 5.17 Run-2: Part program page

5.3 Sample Run 3

In this case the user is assumed to be a registered user and he directly signs into the site with his username and password. The log in page is depicted in Figure 5.18. The new or past project selection page is shown in Figure 5.19. If the user continues with the new project, material selection page is the subsequent page. The material selection page, in which the copper alloy was chosen as the material, is shown in Figure 5.20.

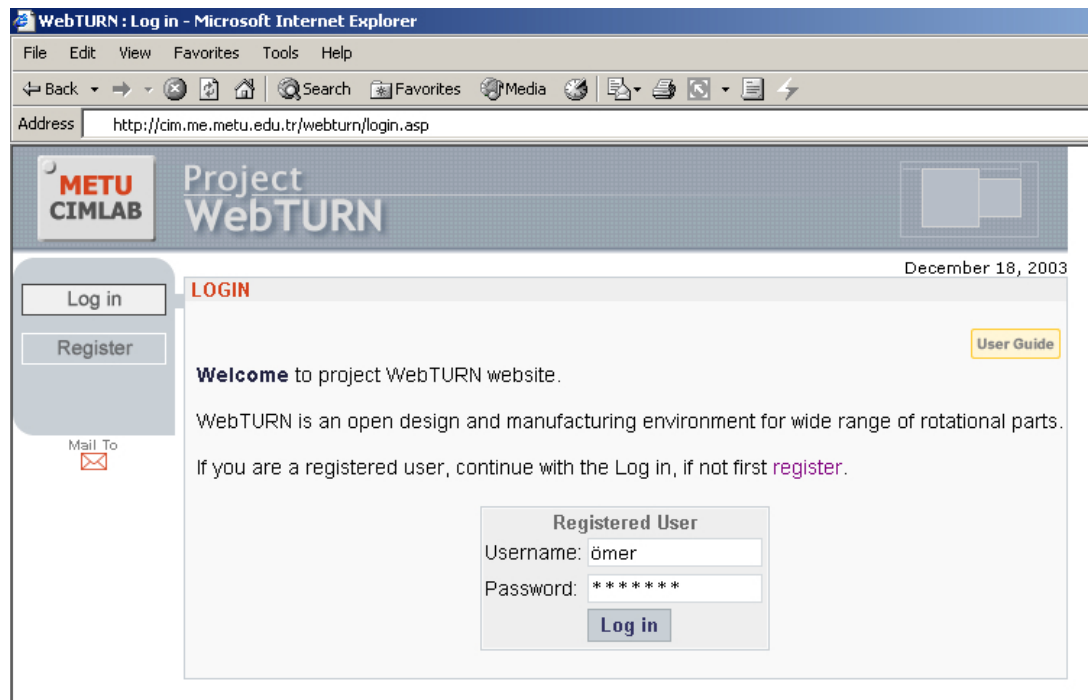


Figure 5.18 Run-3: Log in page

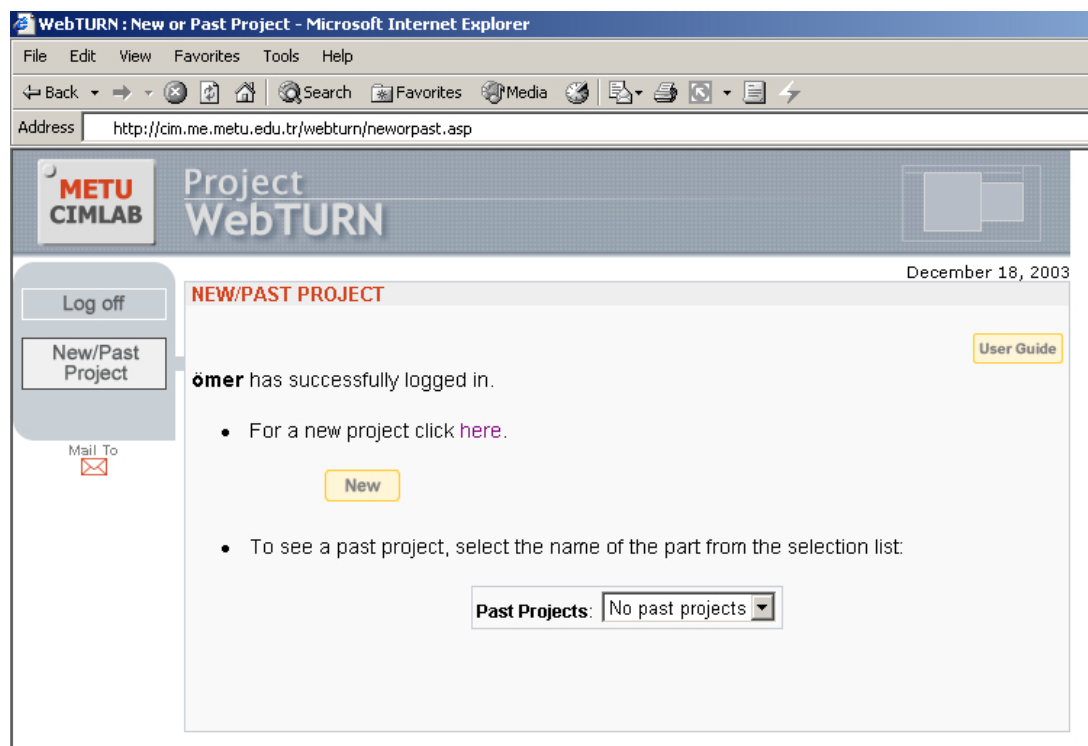


Figure 5.19 Run-3: New or past project selection

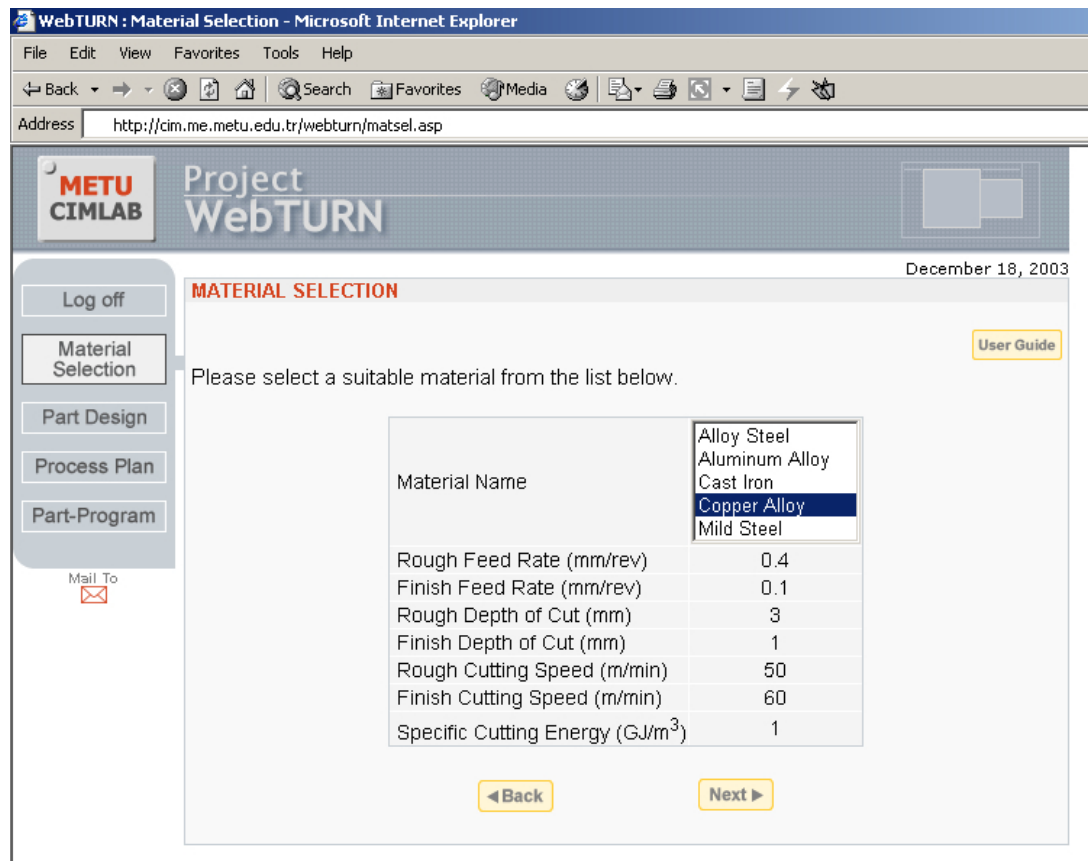


Figure 5.20 Run-3: Material selection

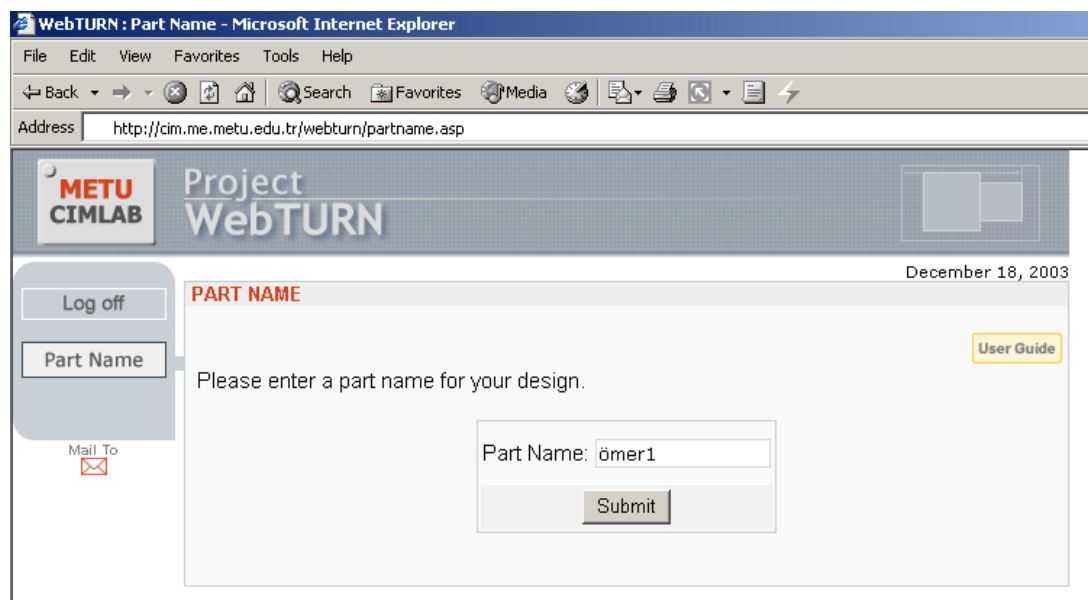


Figure 5.21 Run-3: Part name entry page

The engineering drawing of the part that is going to be sketched is demonstrated in Figure 5.22 and the sketched part is shown in Figure 5.23.

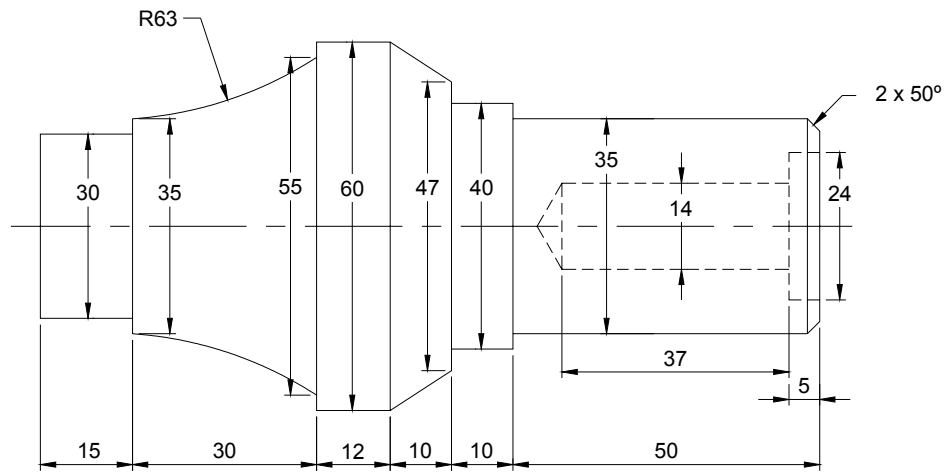


Figure 5.22 Run-3: Engineering drawing of the part (units in mm)

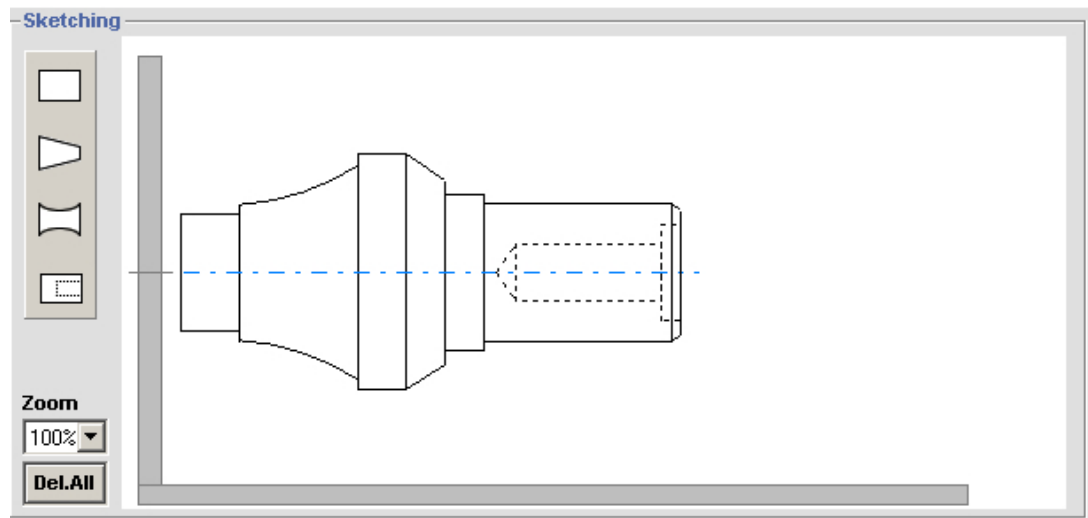


Figure 5.23 Run-3: Sketched part in design stage

The content of the part data file generated for the designed part is shown in Table 5.3.

Table 5.3 Run-3: Part data file

```

"Owner", "Erhan Özsüer"
"Creation Date", "12/18/2003"
"Feature", "Turn"
"Type", "StraightCylinder", 0
"D", 30, 0, 0
"L", 15, 0, 0
"Feature", "Convex&Concave"
"Type", "Concave", 0
"D1", 35, 0, 0
"D2", 55, 0, 0
"L", 30, 0, 0
"R", 63
"CenterCoords", 62.85, 4.28
"Feature", "Turn"
"Type", "StraightCylinder", 0
"D", 60, 0, 0
"L", 12, 0, 0
"Feature", "Taper"
"Type", "Converging_Taper", 0
"D1", 60, 0, 0
"D2", 47, 0, 0
"L", 10, 0, 0
"Feature", "Turn"
"Type", "StraightCylinder", 0
"D", 40, 0, 0
"L", 10, 0, 0
"Feature", "Turn"
"Type", "RChamfer", 0
"D", 35, 0, 0
"L", 50, 0, 0
"RChamfer", 2, 50
"Feature", "Internal"
"Type", "FlatEndDrill"
"D", 24
"L", 5
"Feature", "Internal"
"Type", "ConicalEndDrill"
"D", 14
"L", 37

```

After the part is saved into the database the part, maximum diameter and the total part length are calculated and returned to the web page. By taking into account those values, a billet diameter and a length are recommended on the web page. User may change those values. Also the cutting parameters can be changed on this page just before process planning. Billet definition and cutting parameters edit page is shown in Figure 5.24.

METU CIMLAB Project WebTURN December 18, 2003

Log off Material Selection Part Design **Process Plan** Part-Program

Mail To

PROCESS PLAN [User Guide](#)

Designed Part Specification

Part Name:	ömer1
Material:	Copper Alloy
Maximum Diameter:	60 mm
Total Part Length:	127 mm

Blank Size

Blank Diameter:	70 mm
Blank Length:	177 mm

Cutting Parameters

Rough Feed Rate	0.2	mm/rev
Finish Feed Rate	0.1	mm/rev
Rough Depth of Cut	2	mm
Finish Depth of Cut	1	mm
Rough Cutting Speed	50	m/min
Finish Cutting Speed	60	m/min
Specific Cutting Energy	1	GJ/m ³

[Next ►](#)

Figure 5.24 Run-3: Billet definition and cutting parameters edit page

The generated process plan and part program are shown in Figure 5.25 and 5.26.

WebTURN : Process Plan - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media Print Copy Paste

Address http://cim.me.metu.edu.tr/webturn/processplan2.asp

METU CIMLAB Project WebTURN

December 18, 2003

Log off

Material Selection

Part Design

Process Plan

Part-Program

Mail To

PROCESS PLAN [User Guide](#)

Operation	Tool	Feed rate (mm/rev)	Cutting speed (m/min)	DOC (mm)	Time (min)	Power (Watt)
Facing	T1	0.1	60	1	1.28	118
Rough Right Turning	T1	0.2	50	2	8.36	392
Rough Left Turning	T7	0.2	50	2	4.88	392
Finish Right Turning	T1	0.1	60	1	3.3	118
Finish Left Turning	T7	0.1	60	1	1.89	118
Center Drilling	T2	0.2	50	-	0.04	15
Drilling, D=14	T4	0.2	50	-	0.92	117
Drilling, D=24	T6	0.2	50	-	0.11	343
Parting	T3	0.1	60	2	1.28	235

[Next >](#)

Figure 5.25 Run-3: Process plan page

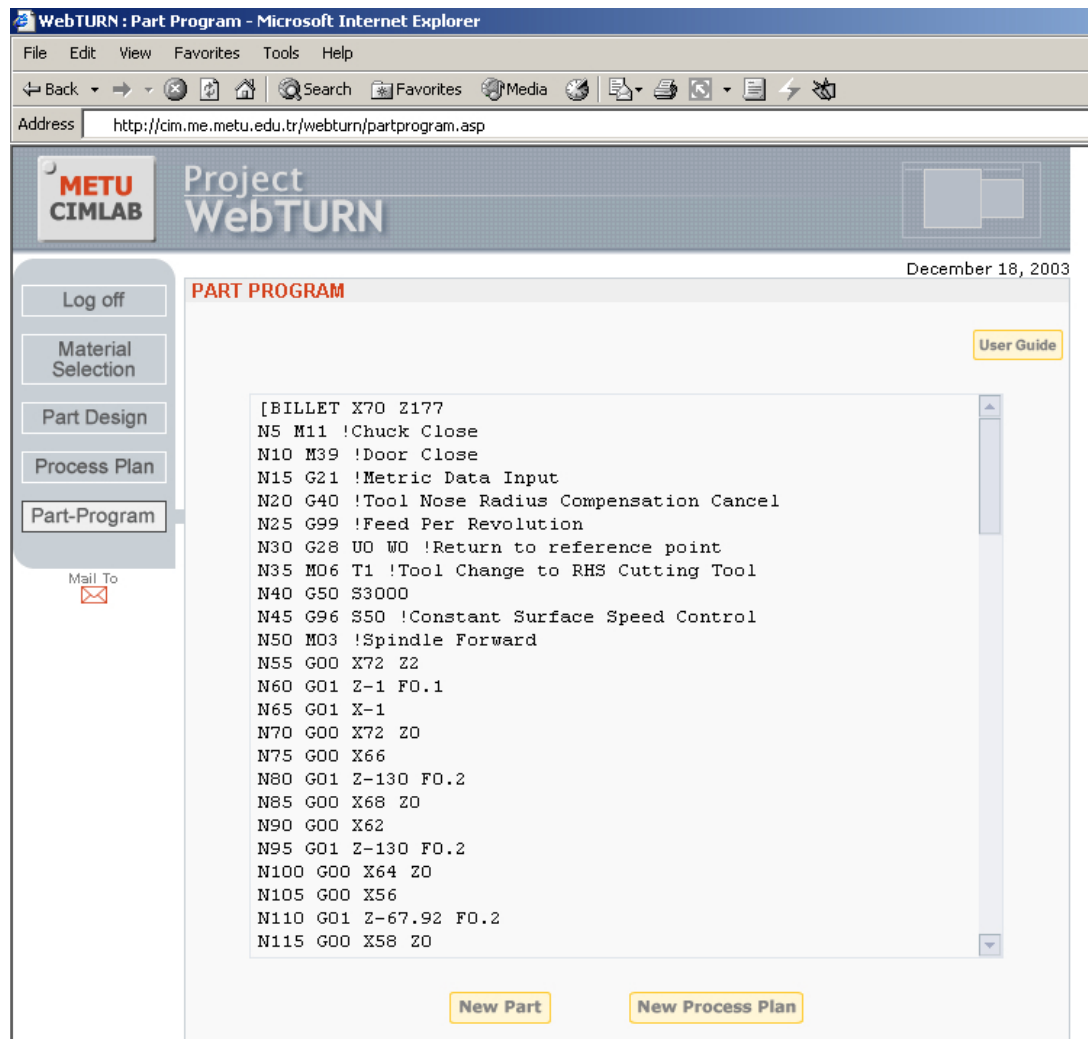


Figure 5.26 Run-3: Part program page

The created projects can be displayed after the part program generation is completed. The complete data of the previous project is show in Figure 5.27.

December 18, 2003

Log off

Past Project

Mail To

PAST PROJECT

[User Guide](#)

Part Name ömer1

Process Plan 1 ▾

Part Material Copper Alloy

Part Drawing

Process Plan

Operation	Tool	Feed rate (mm/rev)	Cutting speed (m/min)	DOC (mm)	Time (min)	Power (Watt)
Facing	T1	0.1	60	1	1.28	118
Rough Right Turning	T1	0.2	50	2	8.36	392
Rough Left Turning	T7	0.2	50	2	4.88	392
Finish Right Turning	T1	0.1	60	1	3.3	118
Finish Left Turning	T7	0.1	60	1	1.89	118
Center Drilling	T2	0.2	50	-	0.04	15
Drilling, D=14	T4	0.2	50	-	0.92	117
Drilling, D=24	T6	0.2	50	-	0.11	343
Parting	T3	0.1	60	2	1.28	235

Part Program

```

[BILLET X70 Z177
N5 M11 !Chuck Close
N10 M39 !Door Close
N15 G21 !Metric Data Input
N20 G40 !Tool Nose Radius Compensation Cancel
N25 G99 !Feed Per Revolution
N30 G28 U0 W0 !Return to reference point
N35 M06 T1 !Tool Change to RHS Cutting Tool
N40 G50 S3000
N45 G96 S50 !Constant Surface Speed Control
N50 M03 !Spindle Forward
N55 G00 X72 Z2
N60 G01 Z-1 F0.1
N65 G01 X-1
N70 G00 X72 Z0

```

◀ Back

New Process Plan

New Part

Figure 5.27 Run-3: Past project view

94

CHAPTER 6

CONCLUSION AND FUTURE WORK

New developments in information technology are changing the instruments of information sharing. World have become smaller than ever with the advent of Internet and other relating technologies. Companies start to make use of Internet and currently many of them have a site on World Wide Web (WWW) with the information about themselves. Being known and reachable by more people has become much more possible with the Internet now. Like others, manufacturing industry starts to take part in this world.

The aim of this study was the development of an interactive and responsive web-based environment for the design and the manufacture of rotational parts. It is called WebTURN.

The previously developed CAD/CAM software package, AutoTURN [19, 47, 48, 49] was a stand-alone application for Windows operating system. It is possible to design a rotational part and to generate the process plan and part program of the part by using this package.

This work shares the idea of developing a CAD/CAM system for rotational parts with the AutoTURN. However, this work is completely a different application with respect to running environment and coding strategy. The web provides the platform independency of the application. In other words, WebTURN runs on any computer which is connected to Internet and having an Internet browser.

In this study, the feature modeler is recreated and clear definitions of design features are provided. A different, easy product data representation format for defined features is generated and used in the application.

In process planning, a different method is used compared to AutoTURN. AutoTURN determines the cutting planes of individual features then combines the estimated cutting planes. Indeed the cutting planes of the features are not independent of each other so those cutting planes should be corrected to coincide with each other. In WebTURN, backward approach is used in process planning which handles the process planning from finish product to blank material. It identifies the whole machining pockets then divides the identified pockets into cutting planes.

In spite of the tight restrictions while developing interactive graphics for the web environment, the real time sketching of the part features become possible.

Although the application is successful in doing many tasks, it needs be improved in various aspects further.

A complete machinability module can be added. New types of material and their properties can be appended to the available material database.

Also the number of machine tools can be increased and related details might be included in the database. Thus, the work would not be specific to a unique machine tool that is Denford CNC lathe.

A separate cutting tool database can be added and present automatic tool selection might be carried out from an extended list.

The necessary cutting parameters required in process planning and part programming might be taken from the study called OPTIM [50], which is a web

based cutting parameters optimization work completed at METU CIM Laboratory. Therefore, the integration of those two studies can be done.

In part design module the diversity of the internal features and sub-features on both external and internal ones can be extended.

Instead of using a system specific product data file representation, STEP (Standard for Exchange of Product model data) that is an ISO standard CAD neutral format, can be used. It is the international standard for exchanging data between different CAD/CAM and Product data Management (PDM) systems. It is independent of any particular system.

An ActiveX component is used in the feature-based modeling. The restrictions in ActiveX control development hinder the development of advanced graphical tools. To overcome those restrictions third party components might be used similar to the ones in this work. Java is a web optimized programming language and includes rich set of APIs (Application Programming Interface) and pre-build classes for computer graphics. Java Applets are the solution of Java for complicated web user interface designs similar to Microsoft's ActiveX technology. Java Applet might be preferred over ActiveX in the development of part design component.

REFERENCES

- [1] Nagalingam S. V., Lin G. C. I., 1999, Latest Developments in CIM, Robotics and Computer Integrated Manufacturing, Vol. 5, pp. 423-430.
- [2] Harington J., 1973, Computer integrated manufacturing, New York: Industrial Press.
- [3] Foston A. L., Smith C. L., Au T., 1991, Fundamentals of computer integrated manufacturing, Englewood Cliffs, NJ: Prentice-Hall.
- [4] Groover M. P., 1987, Automation, production systems, and computer integrated manufacturing. Englewood Cliffs, NJ: Prentice-Hall.
- [5] Hunt D. V., 1987, Dictionary of advanced manufacturing technology, New York: Elsevier Science.
- [6] Tian G. Y., Yin G., Taylor D., 2001, Internet-based manufacturing: A review and a new infrastructure for distributed intelligent manufacturing, Journal of Intelligent Manufacturing, Vol.13, pp. 323-338.
- [7] Syan C., Menon U., 1994, Concurrent engineering: concepts, implementation and practice, London: Chapman & Hall.
- [8] Lamming R., 1993, Beyond partnership: strategies for innovation and lean supply. UK: Prentice-Hall.

- [9] Montgomery C., Levine L. O., 1996, The transition to agile manufacturing, Milwaukee: ASQC Quality Press.
- [10] Distributed Applications with Microsoft Visual Basic 6.0, MSDN Training Kit, 1999, Microsoft Press.
- [11] Microsoft Developers Network (MSDN) Library, July 2001.
- [12] Kriegel A., Microsoft SQL Server 2000 Weekend Crash Course, 2001, Hungry Minds.
- [13] <http://www.w3schools.com>
- [14] http://www.cc.gatech.edu/company/SW_Eng/people/Phd/ce.html
Concurrent engineering resources on the web.
- [15] Cho H, Jung M., 1996, Enabling technologies of agile manufacturing and its related activities in Korea, Computer Integrated Engineering, Vol. 30, pp. 323-334.
- [16] Gunasekaran A., 1999, Agile manufacturing: a framework for research and development. International Journal of Production Economy, Vol. 62, pp. 87-105.
- [17] Shukla C, Vazquez M, Chen FF., 1996, Virtual manufacturing: an overview. Computer Integrated Engineering, Vol. 31(1/2), pp. 79-82.
- [18] Moore P.R., Pu J., Ng H.C., Wong C.B., Chong S.K., Chen X., Adolfsson J., Olofsgard P., Lundgren J.-O., 2003, Virtual engineering: an integrated approach to agile manufacturing machinery design and control, Mechatronics.

- [19] Başsoy A. G., Development of the Process Planning Module for a Desktop CAD/CAM Package for Rotational Components, MS Thesis in Mechanical Engineering Department, Middle East Technical University, Ankara, TURKEY, 1998.
- [20] Kesheng Wang, 1998, An integrated intelligent process planning system (IIPPS) for machining, Journal of Intelligent Manufacturing, vol. 9, pp. 503-514.
- [21] Rao P. N., MARA Institute of Technology, Jasthi S. R. K., Structural Dynamic Research Corporation, Computer-Aided Process Planning (CAPP) in Manufacturing Systems and Their Implementation, 2001, CRC Press LCC.
- [22] Feng S. C., Zhang C. C., A modular architecture for rapid development of CAPP systems for agile manufacturing, IIE Transactions (1998) 30, 893-903.
- [23] <http://www.npd-solutions.com/capp.html>, Kenneth Crow, DRM Associates.
- [24] Chen K.-H., Chen S.-J. (Gray), Lin L. and Changchien S. W., 1998, “An integrated graphical user interface (GUI) for concurrent engineering design of mechanical parts”, Computer Integrated Manufacturing Systems Vol. 11, No. 1-2, pp. 91-112.
- [25] Lynch M., Computer Numerical Control for Machining, 1992, McGraw-hill Inc., USA.
- [26] Virtual Reality (VR) CNC Turning Demo Software, 2000, Denford Ltd.
- [27] Balena F., 1999, Programming with Visual Basic 6.0, Microsoft Press.
- [28] http://www.wikipedia.org/wiki/Object-oriented_programming.

- [29] Pender T. A., 2002, UML Weekend Crash Course, Wiley Publishing Inc.
- [30] Saygın, C., A Rule Based Approach in Sequencing Machining Operations For Rotational Components, MS Thesis in Mechanical Engineering Department, Middle East Technical University, Ankara, TURKEY, 1992.
- [31] Mestçi, H., Sevinç, A., Akkurt, M., 1994, Bilgisayarda CNC Torna için Process Planı Optimizasyonu Yapan Uzman Sistem Yaklaşımıyla Bir CAD-CAM Paketi Hazırlanması”, 6th International Machine Design and Production Conference, Middle East Technical University, Ankara, TURKEY.
- [32] Arslan, E., Söylemez, E., 1994, İki Eksenli Sayısal Denetimli Takım Tezgahları için Bilgisayar Destekli İşlem Planla ve Uygulaması, 6th International Machine Design and Production Conference, Middle East Technical University, Ankara, TURKEY.
- [33] Filiz, I. H., Dereli, T., Kayacan, M. C., 1994. “ROMOD: A Production modeling System for Rotational Parts”, 6th International Machine Design and Production Conference, Middle East Technical University, Ankara, TURKEY.
- [34] Talşık E. İ., Development of a Process Planning Package for Rotational Parts, MS Thesis in Mechanical Engineering Department, Middle East Technical University, Ankara, TURKEY, 2001.
- [35] Phillips R. H., Mouleeswaran C. B., 1985, A knowledge-based approach to generative process planning.
- [36] Houten M., 1986, Strategy in Generative Planning of Turning Processes, Annals of CIRP, Vol. 35/1.
- [37] Guisti F., Santochi M., 1989, KAPLAN: A knowledge-based approach to process planning of rotational parts, Annals of CIRP, Vol. 38/1.

- [38] Boer C. R., Petitti M., Lombardi F., Simon J. P., 1990, A CAPP/CAM expert system for a high productivity, high flexibility CNC turning center, *Annals of CIRP*, Vol. 39/1.
- [39] Cho K. K., LeeS. H., Ahn J. H., Development of Integrated Process Planning and Monitoring System for Turning Operation, *Annals of CIRP*, Vol. 40/1, 1991.
- [40] Yeo S. H. Wong Y. S., Rahman M., 1991, Integrated knowledge-based machining system for rotational parts, *Int. J. Prod. Res.* Vol. 29, No. 7, pp. 1325-1337.
- [41] Younis M. A. and Abdel Wahab M. A., 1997, A CAPP expert system for rotational components, *Computers Ind. Engng.*, Vol. 33, No. 3-4, pp. 509-512.
- [42] Ahn S., Sequin C., Wright P., Computer Science and Mechanical Engineering Departments, University of California Berkeley, Final Report, 1998-1999.
- [43] McMains S., Sequin C., Smith C., Wright P., Computer Science Department and Mechanical Engineering Departments, University of California Berkeley, Final Report, 1999-2000.
- [44] Adamczyk Z., Kociolek K., 2001, CAD/CAM technological environment creation as an interactive application on the Web, *Journal of Materials Processing Technology*, Vol. 109, pp, 222-228.
- [45] Huang G. Q., Lee S. W., Mak K. L., 1999, Web-based product and process data modeling in concurrent “design for X”, *Robotics and Computer-Integrated Manufacturing*, Vol. 15, pp. 53-63.

- [46] Quang L., Zhang Y. F., Nee A. Y. C., 2001, A Distributive and Collaborative Concurrent Product Design System through the WWW/Internet, International Journal Advanced Manufacturing Technology, Vol. 17, pp 315-322.
- [47] İlik Y., A Desktop CAD/CAM Package for Rotational Parts, MS Thesis in Mechanical Engineering Department, Middle East Technical University, Ankara, TURKEY, 1995.
- [48] Öztürk F., Machinability and Tool Databases with Automatic Tool Selection Module for a Desktop Turning CAD/CAM Package, MS Thesis in Mechanical Engineering Department, Middle East Technical University, Ankara, TURKEY, 1998.
- [49] Susuzlu T., NC Verification For Turning Operations, MS Thesis in Mechanical Engineering Department, Middle East Technical University, Ankara, TURKEY, 2001.
- [50] Sarı B., Development of a Job Shop Planning System: Optimization of Machining Operations Using Windows DNA Architecture, Ankara, TURKEY, 2001.

APPENDIX A

USER MANUAL

WebTURN is hosted by a server in METU-CIM Laboratory. The Windows 2000 Professional has been installed on the server and the server name is CIMSTORE. To turn it into a web server the IIS 3.0(Internet Information Service) is installed on it. The website is placed into the default web directory that is “C:\inetpub\wwwroot\webturn”. The URL of the web server is “http://cim.me.metu.edu.tr/”. Microsoft SQL Server 2000 is installed on the server. The system’s complete database is built on it. The address of the application is “http://cim.me.metu.edu.tr/webturn/”. Homepage of the website is shown in Figure A.1.

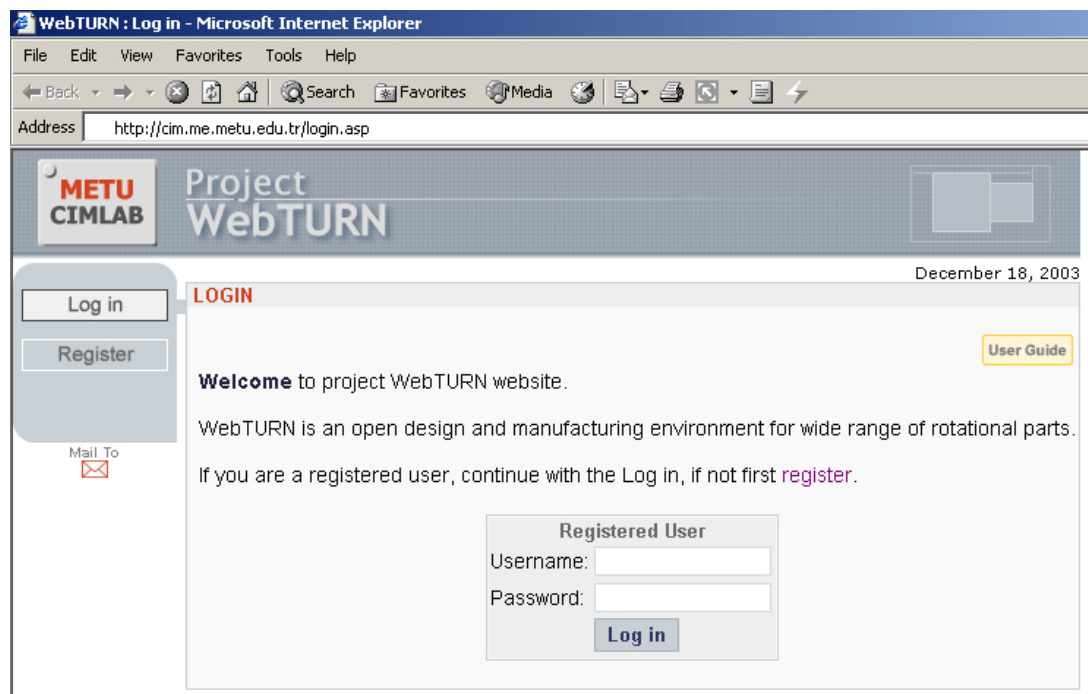


Figure A.1 WebTURN home page

A third party user controller named Flipper CAD Controller 2.5 is used in development of geometric design controller. This controller can be reached from <http://www.proworks.com>.

The coded business classes for part reading, process planning and part programming are compiled in Visual Basic 6 and “WebTURN.dll” file is obtained. This file should be installed into the web server. The installation procedure is depicted in Figures A.2 and A.3.

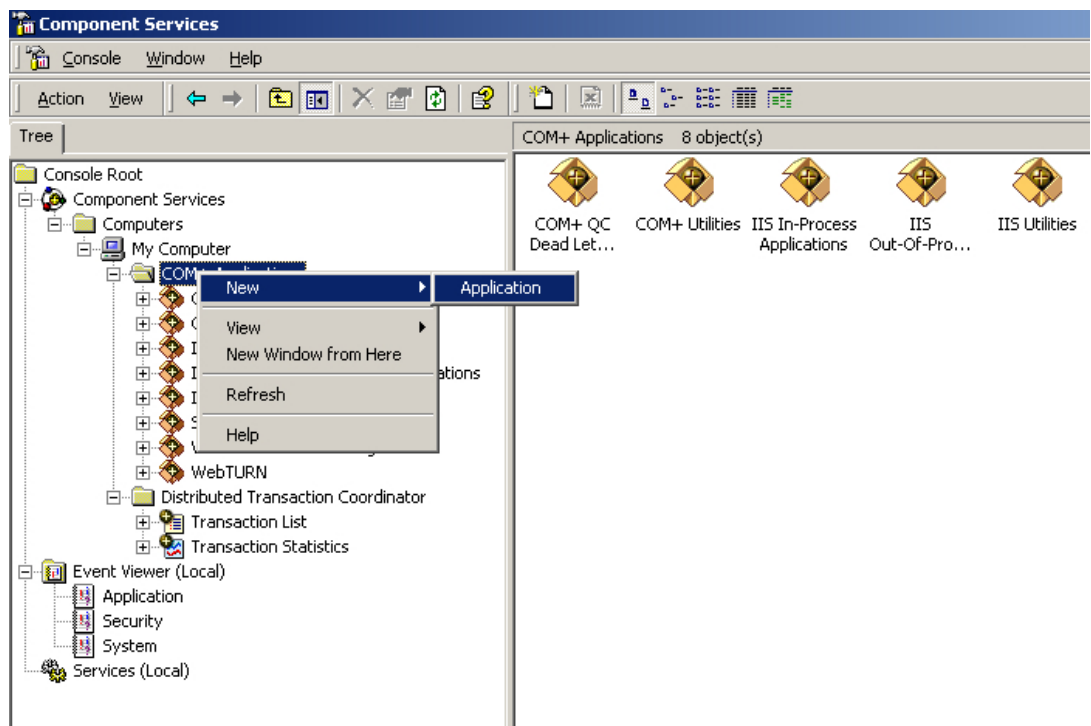


Figure A.2 MS Windows 2000 components services window-1

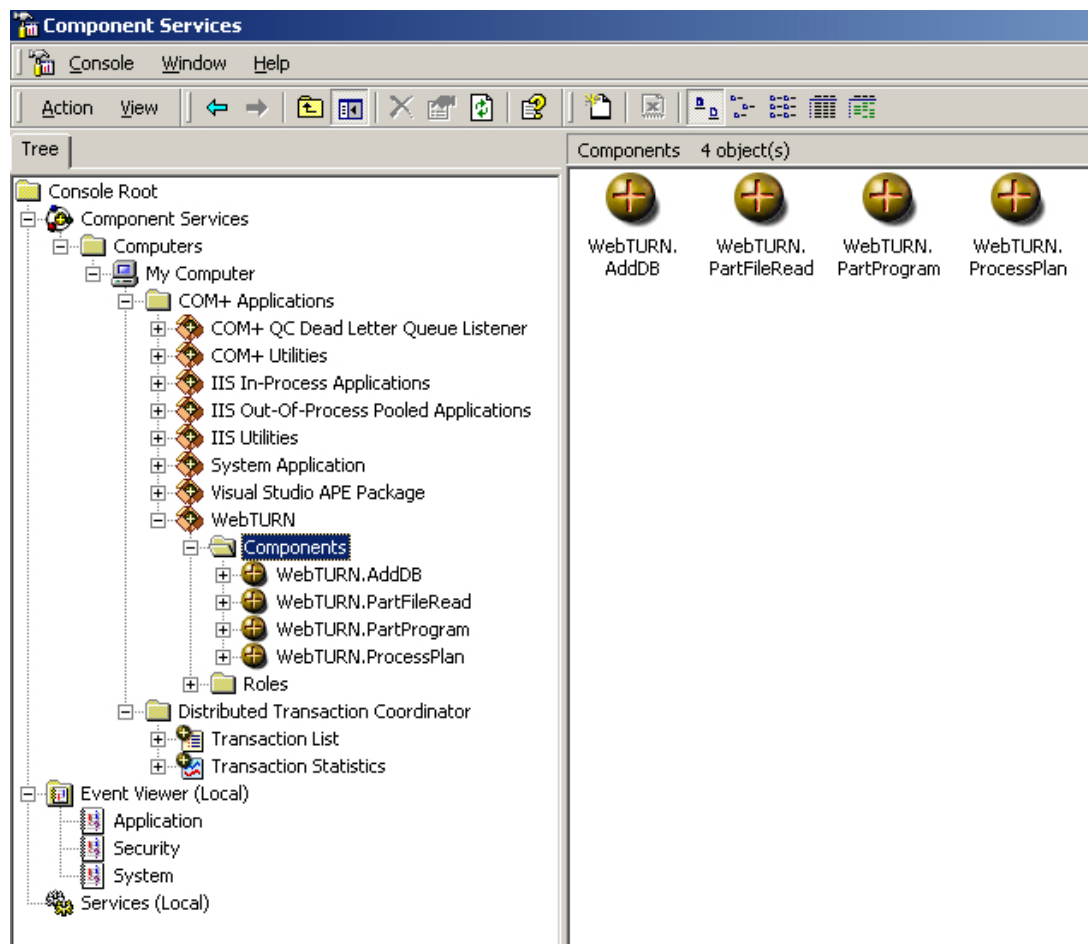


Figure A.3 MS Windows 2000 components services window-2

APPENDIX B

UML DIAGRAMS

B.1 Use Case Diagram

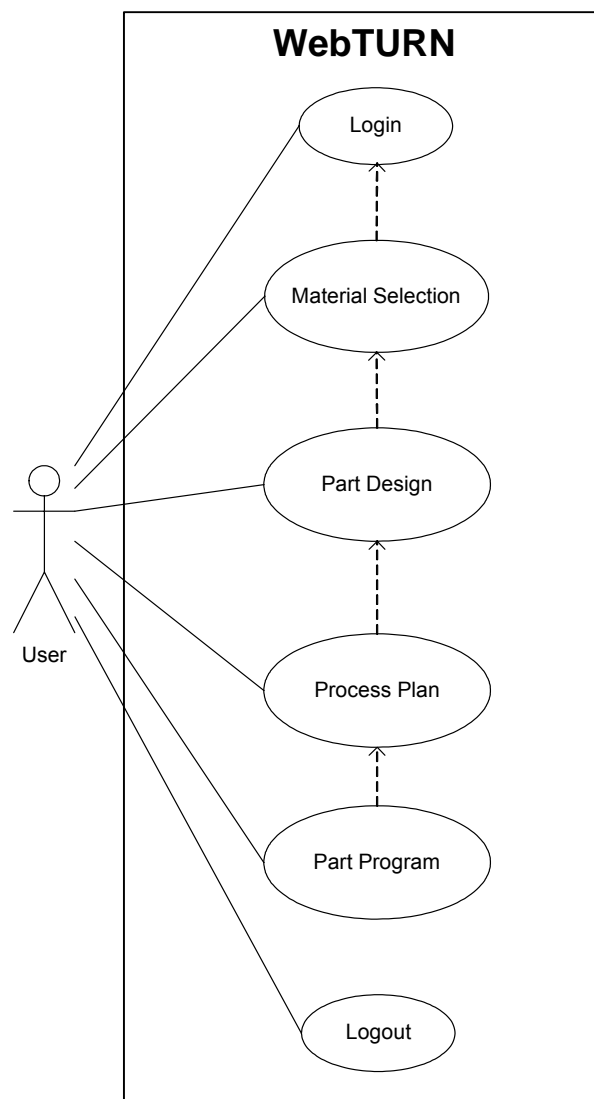


Figure B.1 Use case diagram

B.2 Activity Diagram

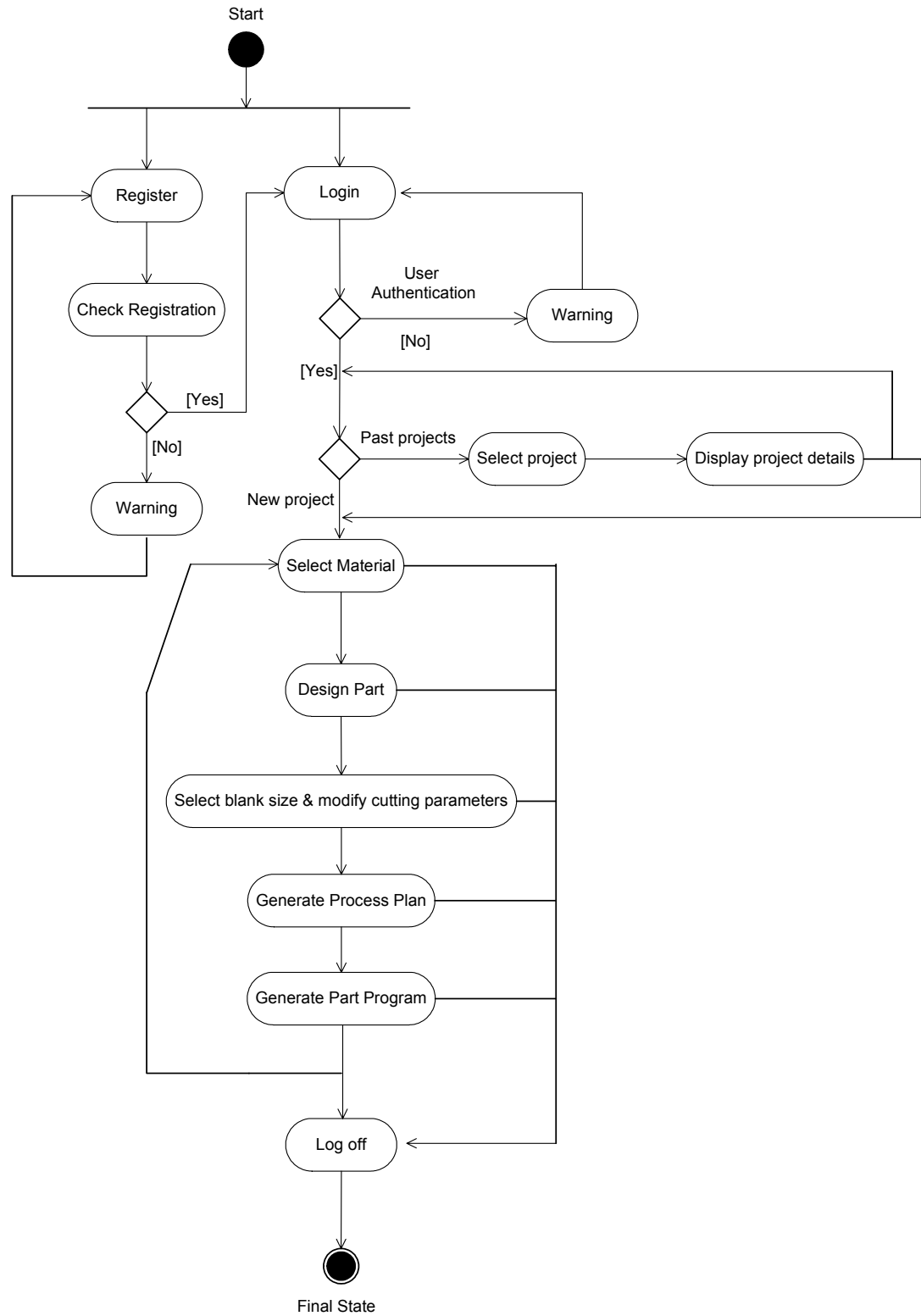


Figure B.2 Activity diagram

APPENDIX C

G CODES

C.1 Sample Run 1

[BILLET X80 Z176	N130 G00 X60 Z0
N5 M11 !Chuck Close	N135 G00 X52
N10 M39 !Door Close	N140 G01 Z-93.5 F0.4
N15 G21 !Metric Data Input	N145 G00 X54 Z0
N20 G40 !Tool Nose Radius Compensation Cancel	N150 G00 X47
N25 G99 !Feed Per Revolution	N155 G01 Z-90 F0.4
N30 G28 U0 W0 !Return to reference point	N160 G00 X49 Z0
N35 M06 T1 !Tool Change to RHS Cutting Tool	N165 G00 X42
N40 G50 S3000	N170 G01 Z-69 F0.4
N45 G96 S50 !Constant Surface Speed Control	N175 G00 X44 Z0
N50 M03 !Spindle Forward	N180 G96 S60
N55 G00 X82 Z2	N185 G00 X37.54
N60 G01 Z-1 F0.1	N190 G01 Z-1 F0.1
N65 G01 X-1	N195 G01 X40 Z-3
N70 G00 X82 Z0	N200 G01 Z-70 F0.1
N75 G00 X77	N205 G01 X45 F0.1
N80 G01 Z-148 F0.4	N210 G01 Z-91 F0.1
N85 G00 X79 Z0	N215 G02 X50 Z-91 R2.5
N90 G00 X72	N220 G01 X70 Z-116
N95 G01 Z-130 F0.4	N225 G01 Z-131 F0.1
N100 G00 X74 Z0	N230 G01 X75 F0.1
N105 G00 X64	N235 G28 U0 W0
N110 G01 Z-93.5 F0.4	N240 M06 T3 !Tool change to grooving tool
N115 G00 X66 Z0	N245 G00 Z-122
N120 G00 X58	N250 G01 X66 F0.1
N125 G01 Z-93.5 F0.4	N255 G01 Z-126

N260 G28 U0 W0
 N265 M06 T5 !Tool change to threading tool
 N270 G00 Z-1
 N275 G00 X42
 N280 G92 X38 Z-65 F1.5
 N285 X36
 N290 G28 U0 W0
 N295 M06 T3 !Tool change to parting-off tool
 N300 G00 Z-148
 N305 G01 X-1 F0.1
 N310 M05 !Spindle stop
 N315 M38 !Door open
 N320 M10 !Chuck open
 N325 M30 !Program reset & rewind

C.2 Sample Run 2

[BILLET X30 Z80	N105 G00 X19
N5 M11 !Chuck Close	N110 G01 Z-26.64 F0.15
N10 M39 !Door Close	N115 G00 X21 Z0
N15 G21 !Metric Data Input	N120 G00 X15
N20 G40 !Tool Nose Radius Compensation Cancel	N125 G01 Z-23 F0.15
N25 G99 !Feed Per Revolution	N130 G00 X17 Z0
N30 G28 U0 W0 !Return to reference point	N135 G96 S200
N35 M06 T1 !Tool Change to RHS Cutting Tool	N140 G00 X10
N40 G50 S3000	N145 G01 Z-1 F0.1
N45 G96 S150 !Constant Surface Speed Control	N150 G01 X14 Z-3
N50 M03 !Spindle Forward	N155 G01 Z-23 F0.1
N55 G00 X32 Z2	N160 G02 X18 Z-25 R2
N60 G01 Z-1 F0.1	N165 G01 X25 Z-40
N65 G01 X-1	N170 G01 Z-50 F0.1
N70 G00 X32 Z0	N175 G28 U0 W0
N75 G00 X26	N180 M06 T3 !Tool change to grooving tool
N80 G01 Z-52 F0.15	N185 G00 Z-46
N85 G00 X28 Z0	N190 G01 X19 F0.1
N90 G00 X21	N195 G01 X29
N95 G01 Z-30.93 F0.15	N200 G28 U0 W0
N100 G00 X23 Z0	N205 M06 T2 !Tool change to centre drill

N210 G00 X0 Z0
 N215 G01 Z-2 F0.1
 N220 G00 Z0
 N225 G28 U0 W0
 N230 M06 T4 !Tool change to drilling tool
 N235 G00 X0 Z0
 N240 G01 Z-31
 N245 G00 Z0
 N250 G28 U0 W0
 N255 M06 T3 !Tool change to parting-off tool
 N260 G00 Z-52
 N265 G01 X-1 F0.1
 N270 M05 !Spindle stop
 N275 M38 !Door open
 N280 M10 !Chuck open
 N285 M30 !Program reset & rewind

C.3 Sample Run 3

[BILLET X70 Z177	N95 G01 Z-130 F0.2
N5 M11 !Chuck Close	N100 G00 X64 Z0
N10 M39 !Door Close	N105 G00 X56
N15 G21 !Metric Data Input	N110 G01 Z-67.92 F0.2
N20 G40 !Tool Nose Radius Compensation Cancel	N115 G00 X58 Z0
N25 G99 !Feed Per Revolution	N120 G00 X52
N30 G28 U0 W0 !Return to reference point	N125 G01 Z-64.85 F0.2
N35 M06 T1 !Tool Change to RHS Cutting Tool	N130 G00 X54 Z0
N40 G50 S3000	N135 G00 X49
N45 G96 S50 !Constant Surface Speed Control	N140 G01 Z-62.54 F0.2
N50 M03 !Spindle Forward	N145 G00 X51 Z0
N55 G00 X72 Z2	N150 G00 X50
N60 G01 Z-1 F0.1	N155 G01 Z-60 F0.2
N65 G01 X-1	N160 G00 X52 Z0
N70 G00 X72 Z0	N165 G00 X42
N75 G00 X66	N170 G01 Z-60 F0.2
N80 G01 Z-130 F0.2	N175 G00 X44 Z0
N85 G00 X68 Z0	N180 G00 X38
N90 G00 X62	N185 G01 Z-50 F0.2

N190 G00 X40 Z0	N360 G00 X49.72
N195 G00 X37	N365 G01 Z-1 F0.1
N200 G01 Z-50 F0.2	N370 G01 X35 Z-3
N205 G00 X39 Z0	N375 G01 Z-51 F0.1
N210 G28 U0 W0	N380 G01 X40 F0.1
N215 M06 T7 S50 !Tool change to LHS tool	N385 G01 Z-61 F0.1
N220 G00 Z-128	N390 G01 X47 F0.1
N225 G00 X62	N395 G01 X60 Z-71
N230 G01 X58 F0.2	N400 G01 Z-83 F0.1
N235 G01 Z-84 F0.2	N405 G28 U0 W0
N240 G00 X60 Z-128	N410 M06 T7 S60 !Tool change to LHS tool
N245 G01 X55 F0.2	N415 G01 X-113 F0.1
N250 G01 Z-84 F0.2	N420 G01 X35 F0.1
N255 G00 X57 Z-128	N425 G03 X55 Z-83 R63
N260 G01 X51 F0.2	N430 G01 X60 F0.1
N265 G01 Z-86.79 F0.2	N435 G28 U0 W0
N270 G00 X53 Z-128	N440 M06 T4 !Tool change to drilling tool
N275 G01 X47 F0.2	N445 G00 X0 Z0
N280 G01 Z-90.63 F0.2	N450 G01 Z-43
N285 G00 X49 Z-128	N455 G00 Z0
N290 G01 X43 F0.2	N460 G28 U0 W0
N295 G01 Z-95.29 F0.2	N465 M06 T6 !Tool change to drilling tool
N300 G00 X45 Z-128	N470 G00 X0 Z0
N305 G01 X39 F0.2	N475 G01 Z-6
N310 G01 Z-101.46 F0.2	N480 G00 Z0
N315 G00 X41 Z-128	N485 G28 U0 W0
N320 G01 X37 F0.2	N490 M06 T3 !Tool change to parting-off tool
N325 G01 Z-105.8 F0.2	N495 G00 Z-130
N330 G00 X39 Z-128	N500 G01 X-1 F0.1
N335 G01 X32 F0.2	N505 M05 !Spindle stop
N340 G01 Z-114 F0.2	N510 M38 !Door open
N345 G00 X34 Z-128	N515 M10 !Chuck open
N350 G28 U0 W0	N520 M30 !Program reset & rewind
N355 M06 T1 S60 !Tool change to RHS tool	

APPENDIX D

DENFORD MIRAC CNC TURNING MACHINE SPECIFICATIONS

Mirac is a CNC bench type turning machine for educational and advanced technical training. Programming can take place on the integral CNC control or away from the machine via off-line software.

Features of the Mirac:

- Designed specifically for education and training
- Manufactured to industrial standards
- Programming with ISO format, incorporating controls such as FANUC
- Capable of cutting resistant materials such as wax, plastics, acrylics, copper, aluminum and steel.
- Links to range of software (e.g. AutoCAD, ProDesktop, Techsoft) via 'QuickCAM turning' software or with 'Lathe CAM Designer' introductory software
- Totally enclosed interlocked guard
- Feed rate and spindle speed override controls
- Optional 8 station programmable turret

Technical specifications of Mirac series:

- Maximum turning diameter: 100 mm
- Maximum turning length: 180 mm
- Spindle bore no.3MT: 21 mm
- Spindle nose A2 type: 42 mm
- Spindle speeds: 0-5000 rpm
- Feedrates: 0-1200 mm/min
- X axis travel: 85 mm

- Z axis travel: 155 mm
- Index time-station/station: 0.65 seconds
- Index time- full revolution: 1.75 seconds
- Weight: 320 kg