Fall 2001

E -commerce for the metal removal industry

Okhyun Ryou
University of New Hampshire, Durham

Recommended Citation

Ryou, Okhyun, "E -commerce for the metal removal industry" (2001). *Doctoral Dissertations*. 43.
https://scholars.unh.edu/dissertation/43

INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

ProQuest Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

E-COMMERCE FOR THE METAL REMOVAL INDUSTRY

by

Okhyun Ryou

B.S.I.E. Seoul National University, 1991
M.S.I.E. Korea Advanced Institute of Science and Technology, 1993

DISSERTATION

Submitted to the University of New Hampshire
in Partial Fulfillment of
the Requirements for the Degree of

Doctor of Philosophy

In

Engineering: Systems Design

September, 2001

UMI Number: 3022963

Bell & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

This dissertation has been examined and approved.

Dissertation Director, Dr. Robert B. Jerard
Professor of Mechanical Engineering

Dr. Barry K. Fussell
Associate Professor of Mechanical Engineering

Dr. May-Win Thein
Assistant Professor of Mechanical Engineering

Dr. Daniel R. Bergeron
Professor of Computer Science

Dr. Linda G. Sprague
Professor of Operations Management

July 24, 2001
Date

DEDICATION

To my mother, Kim, Nan-Young,

Who devotes herself to my brothers and myself,

And to my father, Ryou, Kyung-Su,

Who guides me to do my best at any time.

ACKNOWLEDGEMENTS

I would like to thank the members of my dissertation committee, Dr. Jerard, Dr. Fussell, Dr. Thein, Dr. Bergeron, and Dr. Sprague, who spared their valuable time and improved the quality of my dissertation with their extensive advice. All members were generous and friendly in helping me finish this dissertation.

Special thanks goes to Professor Byung-Kyu Choi at KAIST for introducing me to Professor Jerard who urged me to enroll in the Ph. D program at UNH. Most of all, I would like to give special thanks to Professor Jerard. From the very first moment of our contact, when I could hardly make any conversation in English, to the conclusion of my Ph. D. program, I never forgot his hospitality, assistance and support at any moment.

More people than I can describe have helped me with this undertaking. Matthew Camillieri, pleasantly allowed me to investigate the metal machining practice at the Stone machine company. Without his help, I could not have finished my dissertation in time. Chad Schuyler spent many hours correcting my dissertation draft without uttering any complaints. Jeff, Nate, Brogan, Douglas and Cuneyt, my colleagues in the Design Manufacturing Lab have always been good friends. With their fellowship, my life was always steered from the monotonous. I would like to thank The National Science Foundation (NSF) who supported this dissertation. I would like to give my sincere appreciation to all.

I am really proud of my first son, Jeun, for his decency in understanding his father's situation. Bringing a pleasant mood into my life and giving warm advice, my wife, Yunhi makes me feel like I am the happiest man in the world.

Because I know my parents in Korea have not slept a comfortable night without caring for my well being during my Ph. D. program, I gratefully appreciate them. I know everything is worthless compared to my parents' love for me in overcoming all the experienced difficulties.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ALGORITHMS

LIST OF TABLES

LIST OF SYMBOLS

2D	Two Dimensions
3D	Three Dimensions
API	Application Programming Interface
APT	Automatically Programmed Tools
ARPANET	Advanced Research Projects Agency Network
A_f	Area of the bottom surface created from the unit machining operation
A_s	Average machining setup time
A_t	Average cutting tool change time
A_u	Average workpiece load and unload time
A_w	Area of the wall surface created from the unit machining operation
B2B	Business to Business
B2C	Business to Consumer
B-Rep	Boundary Representation
C	Total cost for fabricating ordered parts
C2C	Consumer to Consumer
CAD	Computer-Aided Design
CAE	Computer-Aided Engineering
CAM	Computer-Aided Manufacturing
CAPP	Computer-Aided Process Planning
CE	Concurrent Engineering
C_{fix_o}	Total fixed cost per order
C_{fix_p}	Total fixed cost per part
CIF	Caltech Intermediate Form
CLDATA	Cutter Location Data file

C_m	Operator rate and machine overhead
C_p	Cost of part post-machining operations
C_w	Cost of workpiece
CSG	Constructive Solid Geometry
D_c	Tool diameter of the current cutting tool
D_r	Tool diameter of the reference cutting tool
DSG	Destructive Solid Geometry
DTD	Document Type Definition
EDI	Electronic (Design) Data Interchange
EDM	Electronic Discharging Machine
E-Mill	Machined part B2B e-Commerce application
FACILE	Fast Associative Clean Interface Language and Environment
FACILE/Design	The design functional part of the FACILE software systems
FACILE/Fabricate	The fabrication functional part of the FACILE software systems
FBD	Feature Based Design
FBM	Feature Based Manufacturing
F_t	Weight factor changing upon the tolerances or the levels of surface finish imposed on the machining feature
GCS	Global Coordinate System
GD & T	Geometric Dimension & Tolerance
HTML	Hyper Text Markup Language
IC	Integrated Circuit
IGES	Initial Graphics Exchange Specification
LCS	Local Coordinate System
MOSIS	Metal Oxide Silicon Implementation Service
MRR	Material Removal Rate
MRSEV	Material Removal Shape Element Volumes
N_b	Part machining batch size

NCML	Numerically Controlled Markup Language or Simply NC Markup Language
N_s	Number of machining setups
N_t (i)	Cutting tool change number for setup i, i=1.. N_s
OAC	Open Architecture Controllers
OOP	Object Oriented Programming
Q	Quantity of machined parts ordered
R_d	Material removal rate for hole machining
R_{dr}	Reference Material removal rate for hole machining
R_f	Bottom surface cutting rate
R_{fr}	Reference bottom surface cutting rate
RFQ	Request For Quota/Quotation
R_r	Material removal rate for rough cutting
R_{sr}	Reference wall surface cutting rate
R_w	Wall surface cutting rate
SFF	Solid Freeform Fabrication
SGML	Standard Generalized Markup Language
SLA	Stereo-Lithography Apparatus
STEP	Standard for the Exchange of Product Model Data, ISO 10303 Part 224
STL	Stereo Lithography
T_c	Total cutting time
T_m	Machining time
T_p	Pre-machining processing time
T_s	Total setup time
T_t	Total tool change time
T_u	Total workpiece setup time (load/unload time)
UML	Unified Modeling Language
UMO	Unit Machining Operation
VLSI	Very Large Scale Integration

V	Volume removed from the current machining operation
VM	Virtual Manufacturing
W3C	World Wide Web Consortium
WCS	World Coordinate System
XML	Extensible Markup Language

ABSTRACT

E-COMMERCE FOR THE METAL REMOVAL INDUSTRY

By

Okhyun Ryou

University of New Hampshire, September, 2001

The popularity of outsourcing fabrication introduces a problem, namely an inevitable loss of data as information is translated from design to fabrication or from one system to another. Unsatisfactory information, delivered to the outsourcing facility, and inefficient communications between design and fabrication certainly cause enormous economic losses from late product delivery or bad product quality. To overcome these data transferring problems and to improve communications between the design and fabrication sides, a design and manufacturing methodology for custom machined parts in E-Commerce is suggested and implemented in this dissertation. This methodology is based on the idea of a "Clean Interface" like the Mead-Conway approach for VLSI chip fabrication [MEAD81].

Essential design information for fabricating parts properly with NC (Numerical Controlled) milling machines is expressed in machining/manufacturing features, fabrication friendly terminologies, and is represented by a new language called NCML (Numerical Control Markup Language). NCML is based on XML (Extensible Markup Language) - the document-processing standard proposed by the World Wide Web Consortium (W3C). NCML is designed to include the minimum requisite information necessary for the manufacturer to produce the product. The designer defines NCML, which overcomes geographical separation between design and manufacturing, and minimizes unnecessary interactions caused from lack of information.

To prove the possibility of custom machine part fabrication and E-Commerce with NCML, three software systems are implemented. These three systems are FACILE/Design, FACILE/Fabricate, and E-Mill. FACILE is a prototype CAD/CAM system developed to verify NCML feasibility as an Electronic

Data Interchange (EDI) format. FACILE/Design is a system based on manufacturing features like holes, contours, and pockets. It can be used to create geometric models, verify the design, and create NCML files. The NCML file is imported by FACILE/Fabricate and turned into G-codes by applying appropriate cutting conditions. Simplified machining simulation and cost estimation tools using NCML inputs are also developed to show some examples of NCML applications that can help design and manufacturing activities. To demonstrate how NCML could be used in a web-based application, an E-Business model called E-Mill has been implemented. E-Mill is a market place for machined parts whose data is encoded in NCML. To make E-Mill a feasible E-Commerce model, two-way communication based on NCML data and the visualization of 3D geometric models in the Virtual Reality Modeling Language (VRML) are equipped with a competitive matchmaking mechanism.

In this dissertation, a whole system based on NCML bridges the gap between design and manufacturing. As a part of the NCML validation process for the new system, the pros and cons of NCML design features are discussed. A system for cost estimation is calibrated and compared to real cutting results for the purpose of validation.

CHPATER 1

INTRODUCTION

This research attempts to answer the following question: "Is it possible to use E-Commerce for the buying and selling of custom machined parts?" Some historical perspective is first required. During the last two decades, there have been significant improvements in the efficiency of the product development process, due in large part to the increased availability of computing power and related software systems. Engineering software such as CAD/CAM/CAPP/CAE systems, whose capabilities are enhanced by the increase in computer power, are now commonly used at each stage of the product design and manufacturing process.

At the same time there has been a trend toward outsourcing of production as a means of achieving additional economies. Removing the overhead costs of production machinery and labor reduces product development cost. With the aid of the latest developments in transportation and communication, the geographical closeness between design and manufacturing facilities becomes less important.

The popularity of outsourcing introduces a new problem, namely an inevitable loss of data as information is translated from one system to another. Unsatisfactory information delivered to the outsourcing facility may cause enormous economic losses from late product delivery or bad product quality.

E-Commerce has become popular with the advent of widespread Internet use. Economical and efficient market places have appeared on the Internet. However, accurate Electronic Data Interchange is a prerequisite for satisfactory business results for both buyers and sellers. E-Commerce is commonly performed between anonymous participants and geographical closeness and conventional business relationships are no longer required for all successful business enterprises.

Fabrication of machined parts via E-Commerce means that there will be separation between the design and the fabrication facilities. The separation may be spatial, temporal or both. The information transferred from the designer to the manufacturer should contain the minimum requisite information for producing the product. Moreover, the information should be understandable to both the design and the fabrication sides of the process. The information must be accurate and complete, and also be efficiently translated from one application to another. There are a myriad of different functions which must use the data, including design, manufacturing, quotation and purchasing.

A design and manufacturing methodology for machined parts in E-Commerce has been suggested in this dissertation. The methodology is based on the idea of a "Clean Interface" like the Mead-Conway approach for VLSI chip fabrication [MEAD81]. This implies a functional separation between the design and fabrication stages of the product development process. In addition to the conceptual approach, software systems have been developed to verify the proposed design and manufacturing processes.

The following tasks have been performed to achieve the goal of this dissertation.

- A machined part fabrication process, which can be applicable to E-Commerce, is designed and proposed.
- As a fundamental component of the proposed E-Commerce part fabrication process, a machined-part data interchange format called "NCML" has been created. This data format is intended to satisfy the previously mentioned requirements of efficient data handling and sufficient contents for fabricating activities.
- Based on a task analysis of the separate roles of buyers and sellers in E-Commerce, two software systems have been developed; one for the designer (FACILE/DESIGN) and one for the fabricator (FACILE/FABRICATOR). These systems are used to prove the feasibility of the proposed methodology, and to show that it is capable of fulfilling the needs of the design and fabrication functions in circumstances where design and manufacturing are separated.
- A prototype E-Commerce website (E-MILL), which implements the business part of the proposed process was implemented. This is devised to show how the proposed methodology would work on the web and how this can help the metal cutting industry.

1.1 The Related Industry and Issues

1.1.1 The Product Design Process

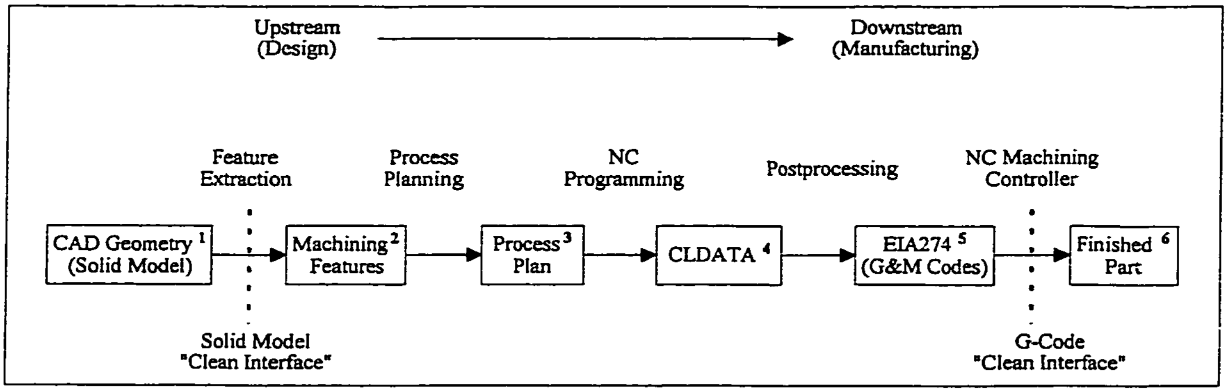


Figure 1.1 The traditional discrete part fabrication process from [JECO98].

The current process for fabricating parts is illustrated in Figure 1.1. In this process the designer creates a geometric model in a CAD system from which machining features are extracted by experienced process planners. The level of detail of the geometric model varies from a simple 2D sketch to a complete solid model depending on the CAD system used and the design requirements. A detailed process plan containing a step-by-step procedure for manufacturing the part is then generated.

The cutter location file (CLDATA) is a non-machine specific format, which defines the required tool paths. In NC programming a skilled programmer uses a CAM package to generate the CLDATA, which is converted into G-codes[1] by a postprocessor for a specific Numerically Controlled (NC) machine. Finally the G-codes are read and executed by the NC machine to achieve the finished part.

While the process shown in Figure 1.1 has worked well for many years, there are at least two potential problems that keep it from being successfully used in an E-commerce environment. First, there are no facilities for communication of upstream intent (design model) to downstream (manufacturing) functions, or for feedback from downstream to upstream to determine cost ramifications of subtle design

---

[1] The G codes are roughly the same on all machines but there are variations and the consequence of incorrect commands could damage the part and the NC machine itself.

changes. Second, each task in the serialized procedure is time consuming and error-prone. Each step can generate different outputs depending on the individual practices of the designer, process planner and machinist. The designer's intent may be misinterpreted leading to unexpected results and a costly rework.

1.1.2 Electronic Design Data Interchange (EDI) Problems

The data describing the part must be provided in a format that is accurate, complete and unambiguous. Furthermore, the usefulness of this information is proportional to the ease with which it can be read and interpreted by machines.

Usually, as seen at the Solid Model "Clean Interface" in Figure 1.1, design information encoded in a specific electronic format is delivered to the fabrication side in the form of geometry. If all designers and fabricators (i.e. buyers and sellers) used only one CAD/CAM software system, we could avoid the problems associated with data interchange. However, even when the same software is used for both design and fabrication, it is not easy to get 100% data interchange due to the different software versions, numerical tolerances, or the internal resource libraries used by different designers and fabricators. In fact, many CAD/CAM software systems are used in industry. Therefore, it is reasonable to assume that the designers and vendors use different CAD/CAM systems, which therefore requires a standard data format for data interchange.

There are many problems related to the data exchange format but probably data loss is the most serious one. It is common to lose some information about the part description due to the different data representation schemes of different data interchange formats. CAD/CAM software has special features and special geometric representations which cannot be expressed in the common standard formats like IGES[2]. Sometimes the loss of as little as 1% of the data can result in useless part files.

"Data repair", which is common in today's environment, would make it impossible to effectively conduct required activities in the follow-up process. To address the Electronic Data Interchange (EDI) problem, many efforts have been made to develop vendor-independent interchange formats.

Another possible problem is that industries still rely on non-electronic data interchange methods

---

[2] An ASME/ANSI standard for the exchange of CAD data

(e.g. engineering drawings). Electronic data often lacks essential part information like tolerances or material specifications, and engineering drawings must be used to augment it. Engineering drawings are certainly a time-honored method of data exchange. They are also often incomplete, inaccurate, and ambiguous. In addition, they are not easily read nor interpreted by machines. To achieve smooth EDI methodology, there is a need to develop another general standard that will include more than geometric information that can be handled in an automated manner.

1.1.3 The Metal Removal Industry

Consider the people who design and manufacture parts produced through the process illustrated in Figure 1.1. The buyers of these types of parts are dispersed in many different industries, e.g. consumer goods, automotive, aerospace, and shipbuilding. The sellers of machining services are often given the title of "Job Shops" or "Machine Shops" and will be referred to as such in this dissertation.

Job Shops usually fabricate parts according to the customer's specifications, typically defined by engineering drawings. The parts vary from order to order, and the procedures change according to the part being fabricated. The following statistics defining the Job Shop industry come from the National Tooling and Machining Association (NTMA, 1999)

- $25 Billion annual revenue
- Critical to country's economic health - create tooling, dies, mold and precision machined parts
- 14000 individual companies with an average of 25-30 employees
- 125,000 skilled machinists with annual capacity of 265 million man-hours (NTMA members only)
- Widespread use of CNC machines
- Use a variety of machine tools to precise shapes and dimensions

*Source: (National tooling and machining association, 1999)

As seen in the previous statistics, the Job Shop industry plays a vital role in the process of getting products to the marketplace. Job shops have fairly simple business activities as follows:

- When a RFQ (Request for Quotation) from the part designer or the buyer is received, the job shop must first decide if it is capable of providing the services to fabricate the parts. This decision may depend on the availability of resources like skilled machinists or the correct type of machinery.
- The job shop must then analyze the supplied part information and develop a quotation.

- If the designer or the buyer accepts the bid on the RFQ, a contract will be made between the designer and the job shop.
- According to the part requirements, the job shops fabricate the ordered parts by following a fabrication process similar to one shown in Figure 1.1.

Since the job shop industry strongly relies on the local design companies for business and on its skilled machinist employees to make the parts, job shops are currently hampered by the following factors:

- Insufficient supply of skilled machinists.
- Uneven workload caused by the business fluctuation of customers.
- Low level of new technology due to small business environment.

1.2 Motivation for New Methods

In the previous section, intrinsic problems in the metal removal industry related to process, EDI and business environments are explained briefly. In this section, the basic approach to resolve the problems is described.

1.2.1 E-Commerce - Enable Small and Medium Size Manufacturers to Compete

There are many reasons why E-Commerce is attractive for solving the problems facing job shops described in Section 1.1.3, and why the metal removal industry should focus on E-Commerce.

- Network Effect[3]. The ease of access (with personal computers) to the market place (web sites) can bring more opportunities for the business.
- Anonymous Competition. The E-Commerce environment provides a market place where small and medium sized fabricators can compete against larger fabricators. To date, most E-business has been conducted between anonymous participants or qualified members of the E-Commerce site. As long as

---

[3] When a market's attractiveness greatly increases with additional buyers and sellers. Additional buyers attract additional sellers and vice versa. This effect has eBay (www.eBay.com) atop the C2C domain.

fabricators are capable of producing parts, the size of the company may not be a criterion for choosing the fabricator.

- Revenue Improvement. A well-defined web-based business mechanism may lead to an efficient E-Commerce model for job shop business activities. This approach could help both the designer and the fabricator; for designers (buyers), this would lead to access to a larger number of suppliers with a variety of capabilities; for fabricators (suppliers) it could lead to a more consistent workload.

1.2.2 Clean Interface Mechanism

Currently, there are two possible break points between designer and fabricator illustrated by the "Clean interface"[4] in Figure 1.1. In the "solid model" clean interface the information rich CAD geometry is sent to the remote fabrication facilities. Although this method has been used as the most typical way of transferring design data to manufacturing, the follow-up downstream tasks cannot be fully automated. It imposes too much burden on the fabrication side in addition to the possible misunderstanding of the design intent. Another problem with this approach is that by totally relieving the designer of responsibility for considering fabrication methods, it also creates a distinct possibility that the part cannot be produced.

The other possible approach is indicated as the "G-Code Clean Interface" in Figure 1.1. In contrast to the Solid Model approach, it imposes too much burden on the design side. The designer has to decide all machining conditions like feed rate, spindle speed and cutting tools. To make things worse, there is practically no flexibility for fabricators to improve machining efficiency in this approach.

In the Mead-Conway approach[5] of VLSI chip fabrication, the chip design information guided by fabrication rules is sent to the fabrication facility in a machine-understandable data format. The chip design information is read and checked by fabrication devices and is directly used to execute the manufacturing operations by the IC chip fabrication device. An analogous approach may be possible in the metal removal industry.

---

[4] The clean interface between design and manufacturing originated from the 1980's Mead-Conway approach.

[5] See section 2.1 for details.

With the advent of Open Architecture Controllers (OAC) [6][Altintas94, MDSI] on modern machine tools, there is no reason that higher-level languages cannot replace G-Codes. As many machine tools already have built-in “macros” to perform basic drilling and pocketing operations, it is not difficult to imagine using the part description and the computational capabilities of the OAC to generate toolpaths right at the milling machine. The Machining/Manufacturing Feature[7] concept can be adopted for this purpose because machining features like holes, slots, and pockets, can be obtained from the design shape and can be processed in a fairly automated manner to generate toolpaths.

A review of the design and fabrication process indicates that there might be a better way to interface between designers and fabricators, which satisfies the following guidelines:

- It should include more manufacturing information than the “Solid Clean Interface”. It should contain enough information about design intent such that it can guide the overall fabrication processes. Machining/manufacturing features can be a good starting point for the new Clean Interface.
- The communication channels between design and manufacturing should include a bi-directional communication channel. Either the designer or fabricator should be able to send information to the other.

1.3 Current Trends

In this section, some noticeable business and research activities related to the metal removal industry are given. This discussion gives some ideas about overall trends in E-Commerce and the current level of implementations applied in the industry.

1.3.1 Business to Business E-Commerce

The Internet has been changing the nature of commerce dramatically for several years and will

---

[6] Conventional machine tool controllers are 'closed' to users who wish to either develop customized functions or expand the capabilities of their machines. An open-architecture controller, on the other hand, provides machine tool manufacturers and users with a readily programmable and expandable platform for developing control systems for their machines

[7] See section 2.2 and 2.3 for details about “Feature” used in the design and the manufacturing.

change the nature of business for many industries, including the job shop industry. While most E-Commerce sites are in the "Consumer to Consumer"(C2C, e.g. Ebay.com) or "Business to Consumer"(B2C, e.g. Amazon.com) domains, the focus is rapidly shifting to the "Business to Business"(B2B) domain which is thought to have a larger potential market than either C2C or B2C [FR2000].

Since most E-Commerce sites, including B2B sites, generally handle standardized goods, products manufactured by job shops do not fall into this category and therefore present a relatively high barrier for entry into E-Commerce. However, E-Commerce in the metal removal industry has the potential to solve chronic uneven workload in the job shop. The current trends in E-Commerce in the metal removal industry are presented in chapter 2.

For successful transactions in E-Commerce, the communication between buyers and sellers is very important. The information must be transferred through the Internet and should be understandable by both sides and be handled in an automated way. The reason that current E-Commerce sites only provide simple and general services and are incapable of carrying out main job shop activities such as custom machined part fabrication commerce is related to the difficulty of describing the design information in a standard format understandable to both sides.

1.3.2 Internet Based design and manufacturing

Internet based design and manufacturing is not a current reality. The Internet can provide a single working environment for all users as long as they have computers connected to the network. In the product design and manufacturing domain on the Internet, Web based design and manufacturing software systems have appeared. As an example of current trends Alibre, Inc. (www.alibre.com) is the first commercial web site to offer an Internet based CAD service providing a remotely hosted interactive design environment for data sharing among engineering and design teams. On April 4, 2000, Alibre announced a partnership with SupplierMarket.com (www.SupplierMarket.com) that intends to integrate the product purchasing process with their web based CAD system. Since this is a very new venture it remains to be seen how effective it is, but it seems very likely that major CAD vendors like CATIA and Pro/E are contemplating similar ventures.

1.3.3 Virtual Manufacturing

In design and manufacturing processes, Virtual Manufacturing (VM) activities are prevalent. VM uses computer models and simulations of the manufacturing processes to aid in the design and production of manufactured products. VM activities attempt to mimic manufacturing processes in a computer to prevent costly rework. Several collaborative systems have been developed to share design information and to check manufacturability for workers remotely located from each other.

Simulating manufacturing processes and validating the design demands in a computer are generally possible in any manufacturing domain and have been a normal task in design. Design checking by knowledge based systems, and cutting simulation/verification are some examples of VM activities in the metal cutting industry.

VM research activities include CyberCut [WW98], a system that provides design checking capability for feasibility at the machining part design stage by applying simple fabrication rules like the required minimum machining corner radius. Many CAM systems (MasterCAM, FeatureCAM, etc) and integrated CAD/CAM systems (CATIA, and Pro/E) provide cutting simulation or verification functions based on the created tool paths. The software systems from Tecnomatix Technology Ltd[8] provide a variety of VM solutions for various manufacturing activities including part assembly, and tolerance analysis.

1.4 Design Goals

The fundamental goal of this research is to provide a clean and smooth interface between design and manufacturing for machined parts. Based on the "clean interface" paradigm inspired by the "VLSI Mead-Conway[9]" approach, the primary task is to demonstrate the possibility of E-Commerce in the machining industry. To validate the new methodology, complete design and manufacturing processes should be tested.

---

[8] Tecnomatix Technology Ltd. (www.tecnomatix.com)

[9] See Section 2.1.

1.4.1 Prospects for the Future system

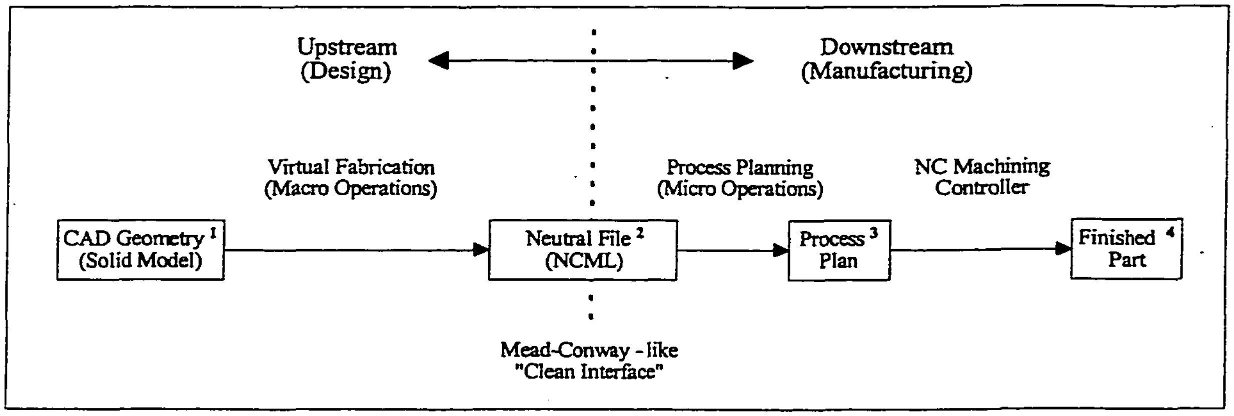


Figure 1.2 The proposed part fabrication process.

Discrete parts, which can be fabricated by a 3-axis NC machining center, are the focus of this research. In contrast to the design and manufacturing processes depicted in Figure 1.1, the methodology pursued in this research is shown in Figure 1.2.

In the design stage, the part design (numbered 1 in Figure 1.2) should be validated to determine if it can be fabricated. This process is indicated as “Virtual Fabrication (Macro simulation)”. The VM concept can be adopted to check machining feasibility by mimicking machining operations in a very simplified manner on the design side. This may be done with general fabrication guidelines or guidelines delivered from the fabricators about the manufacturing feasibility.

After part design validation, the part information is turned into a part data interchange format that contains all necessary information for the fabricator to process tasks without significant additional effort. Bi-directional communication channels are capable of resolving uncertain design related problems and fabricator’s feedbacks. The fabricator will quickly produce G-Codes and execute them on an NC machine to produce the finished part.

At the core of the process is a new part interchange format. In addition to the geometric definition, the minimal required information for smooth downstream operations is included in the part interchange format. Any necessary manufacturing process is carried out with this part interchange format in a fairly automated manner. Therefore, the contents and format of the part interchange, as well as the efficiency of its handling, are important factors affecting the success of this new approach.

The proposed methodology is predicated on the belief that the roles of the designer and fabricator will be spatially and temporally separated, consistent with the E-Commerce environment.

1.4.2 Tasks

The specific goals of this research will be achieved through the following steps.

- Analyze both designers' and fabricators' needs in design and manufacturing. Based on the functional analysis, determine which tasks should be executed. The results of this task can be used for input to software specification.
- Decide what information should be sent from the designer to the fabricator across the clean interface. The result will be the specifications of the new part data interchange format.
- Implementation. At least three software systems need to be developed. To validate all processes in Figure 1.2 and prove the feasibility, all supporting software systems need to be developed. These include a design side software system which covers everything from creating a part design to creating a part data interchange file, and a fabrication side software system which covers everything from importing the transferred part data interchange file, through process planning, to generating G-codes. To demonstrate the possibility of E-Commerce and communication based on the proposed data interchange format, an appropriate E-Commerce application should be modeled and developed.

1.4.3 System Requirements and Criteria for Validation

There will be several system requirements in various categories. The discussions here will be used to define the requirements and guiding principles in this research. The following aspects should be considered.

- The fabricated parts have to be achieved through the proposed processes.
- To show the process improvements, software components should support all processes possible to be automated.
- Software systems should be easy to use by engineers, not by specialized researchers in this area.

As a media for the part data interchange, the new EDI format should satisfy the following requirements.

- Expressive ability. The EDI format should be information-rich data shared by the designer and fabricator for the specific applied area.
- Extensibility. Extending EDI contents should be easy; to extend to other fabrication processes or more complicated geometry should be easily accomplished.
- Web compatibility. For the upcoming business on the web, the EDI format should be compatible with web usage.

Concerning the E-Commerce model, the proposed model should help job shops survive and continue their business in the radically changing business environment. The E-Commerce model should:

- Improve Job Shops productivity and amplify the skilled machinist's capabilities;
- Provide an efficient matchmaking method that allows Job Shops to maintain full utilization of their manpower and machines;
- Provide information that is accurate, complete, unambiguous and easily read and interpreted by machines thereby reducing the Job Shop's burden of quotation and translation of the design model data in the CAM system; and
- Build a communication method between buyers and sellers to allow discussion of design problems and remove ambiguities.

1.5 Overview of Dissertation

The remainder of this dissertation is divided into seven chapters and appendices.

- Chapter 2 discusses relevant literature. Similar approaches in different areas are investigated. Feature based design and manufacturing is briefly reviewed to illustrate its generic simplicity for conversion into manufacturing information. Previous work done in data interchange is reviewed in addition to new web-based approaches.
- Chapter 3 shows the overall software architecture of the developed systems.

- Chapter 4 describes the new part interchange format. The structures and special features in its design are presented.
- Chapter 5 demonstrates the E-Commerce capable applications developed in this research. FACILE-Design, FACILE-Fabricate and E-Mill are explained in detail. Some implementation issues and theoretical background is also given in this chapter. Basic functions and data flow between software systems are also illustrated.
- Chapter 6 describes the test methods for the proposed methodology and software systems.
- Chapter 7 shows the test procedure and results according to methods outlined in Chapter 6.
- Chapter 8 gives conclusions and recommendations for future work.
- APPENDICES provide documents related to the part interchange format specifications, data structures of developed software systems and machined part examples.

CHAPTER 2

RELATED WORK

The work in this research project covers the design and fabrication process, the introduction of a machined part description format, and the development of a relevant software system. Hence, a number of the related software systems, design and fabrication methodologies are discussed in this chapter.

The concept of a clean interface between design and manufacturing is based on the Mead-Conway approach developed for VLSI chip design/fabrication. To apply the Mead-Conway approach in the machining domain, it must be determined how the domain specific information can be delivered. Manufacturing features are one of the feasible concepts for communicating between design and manufacturing. Other successful document standards are reviewed; especially those that would be feasible with other web-based standards and technologies in E-Commerce.

The following items are discussed in this chapter:

- The Mead-Conway approach used in VLSI chip fabrication.
- Feature based approaches either in communicating between design and manufacturing (data exchange) or in implementing software systems.
- Currently available Internet based design and fabrication approaches
- Benchmarking of other successful standard formats.

2.1 The VLSI MOSIS Clean Interface Approach

Mead and Conway [MEAD80, MEAD81, NRC99] developed a simplified, standardized system design methodology and layout design rules for VLSI system and circuit design. Their design methods allowed integrated circuit (IC) designers to quickly and easily design new ICs. Conway also created an innovative new network-based, fast-turnaround VLSI prototyping service. The service enabled chip designers at many locations around the country to submit design files over the Advanced Research Projects

Agency Network (ARPANET) for low cost, rapid fabrication. This system became the basis of what was later called the Metal Oxide Silicon Implementation Service, or MOSIS. It allows students and faculty at Universities to participate in the "silicon revolution" even though they do not have their own silicon fabrication facility. Of course, the very high cost associated with building such a facility is prohibitive for any education institution. The progress achieved by the establishment of this approach is profound, and due to this approach a whole generation of students and faculty are receiving invaluable experience.

Figure 2.1 shows the information flows and processes between MOSIS and chip designers. Circuits need only to be described in a standard format, CIF[1](Caltech Intermediate Form, [MEAD80, Section 4.5]) readable by the MOSIS facilities, and designed in accordance with limitations implicit in the Mead-Conway methodology.

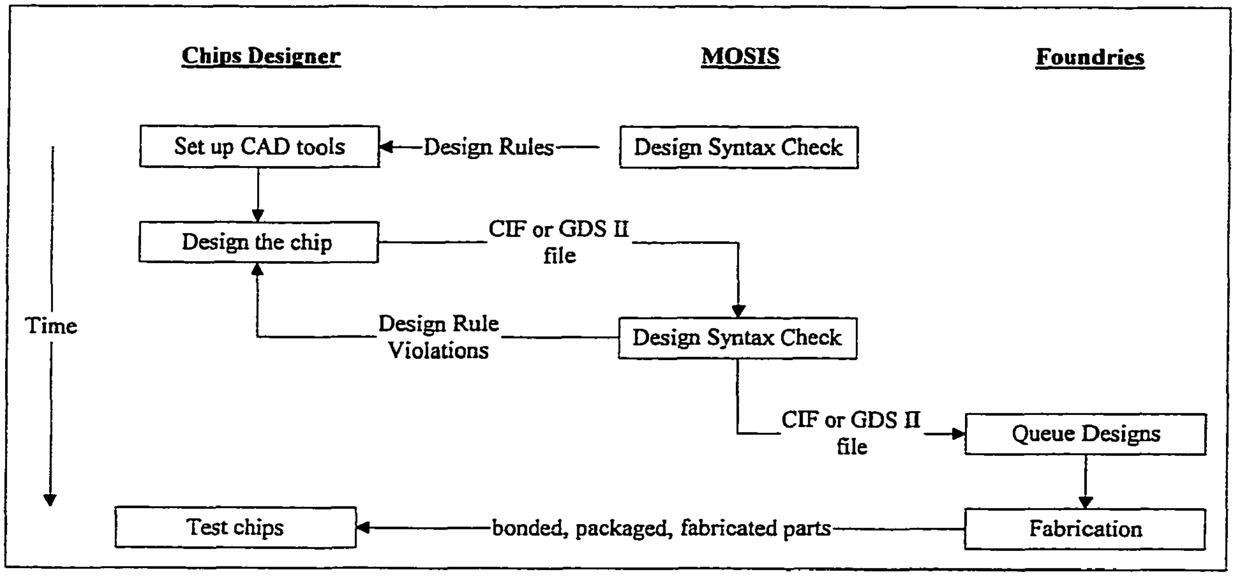


Figure 2.1 MOSIS process flow

Even though this approach was applied in the VLSI circuit fabrication, the experience provides guidance and inspiration for our research. Specifically:

- A Clean Interface is achieved through the data format CIF in which the designer is forced to obey design rules, which affect manufacturing feasibility.

---

[1] Currently, GDS II [RUBIN87, Appendix C] format is also allowed by MOSIS.

- The delivered information is readable by the MOSIS facility-production device in a fairly automated way on the fabrication side.
- It seems possible that a similar process flow approach can be implemented in an E-Commerce environment.

An analogous approach has been implemented for the metal removal industry in this research.

2.2 Feature Based Approach in Design and Manufacturing

To minimize costly and iterative revisions in the product development process, Concurrent Engineering (CE) [CT90, CT92, KKKKO98, SW96] has been adopted. One part of CE is that manufacturing engineers are involved in the process as early as feasible. This is in contrast with a serial approach in which manufacturing does not become involved until design engineers have pretty much finished. Design products, which are either impossible or too expensive to manufacture, are a sure recipe for economic failure. As a core component in CE, feature based approaches have been used to deliver design information for manufacturability analysis. This means that features in the specific domain have been used as information media in CE activities.

Since features represent domain-specific classes of objects, the same form features (geometric shapes) can be interpreted differently depending on the applied domain. Moreover, most old CAD systems mainly concentrate on the geometric aspect of a model; therefore there must be an interpretation process for obtaining features from a design model[2].

The domain-specific feature model[3] is limited to, and guided by the specific engineering activities associated with that domain. If the manufacturing feature model contains enough information, engineers

---

[2] There are three primary approaches for obtaining features from a CAD model. In human-supervised feature recognition, a human user examines an existing CAD model to determine what the manufacturing features are. In automatic feature recognition, the same feature recognition task is performed by a computer system. In design by features, the designer specifies the initial CAD model in terms of various form features, which translate directly into the relevant manufacturing features [SMN94].

[3] A feature model [SMN94] is a collection of feature instances that provide a description of a design with respect to some manufacturing domain.

can carry out manufacturability analyses or may be able to directly produce the part.

The results of such feature based analysis can potentially be used for two purposes in concurrent engineering: (1) to give the production engineer information about what processes and process parameters are most desirable and (2) give the product designer a better understanding of whether and how the design might be changed to improve its manufacturability. For a comprehensive overview of feature-based manufacturing techniques, the reader is referred to [SHMA95, GDRN97].

A significant amount of work has been directed towards defining sets of form features to serve as a communication medium between design and manufacturing [GINDY89, KRAMER92, SHMA95]. For the metal cutting domain, most researchers agree that volumetric features are preferable to surface features, although certain additional information about the surfaces is needed (for example, determining accessibility and tool approach directions). The volumetric features are mapped into machining/manufacturing features. Holes, slots, and pockets in Kramer's machinable feature class [KRAMER92]. This work provides a good example of the process of mapping geometric shapes into domain specific features.

While there is some disagreement[4] about which approach is preferable for practical application, deriving a feature model from the design model can be done in a fairly automated fashion. Nau, Karinthi, Gupta and Regli [GRN95, KN92] have done research on obtaining a feature model from the geometric model, which considers feature relations (interactions between volumetric features) and machining feasibility.

The design by manufacturing (or machining) feature is another automated approach to get a feature model for the downstream activities. This is done by modeling the design with the domain specific form features, e.g. holes, contours, or pockets. Since the design is comprised of features, the design model can be directly used as a fabrication model for process planning, or machining operations. The work of Laakko and Männtylä [LAMä93] couples feature-based design and feature recognition to provide for incremental feature recognition.

---

[4] Two automated methods have pros and cons compared to each other. In the design by manufacturing features approach, infeasible shapes can't be defined since the user is constrained by the choice of operations or features. This methodology also requires that the designer should have some knowledge in manufacturing to understand the usage of features in the design system. The feature recognition approach can generate multiple feature models. However, usually it does not retain tolerance information.

As a more manufacturing centered approach in the Feature Based Manufacturing (FBM), Gaines [GAINES99] developed a tool-centric feature recognizer in the milled part domain. With equipment libraries of machining tools and cutting tools, his system CUSTOM-Cut generates machining volumes that can be machined by the custom shape-cutting tool.

With the advent of parametric design technology[5] and the use of an open architecture solid modeling kernel[6], design by feature is likely to be integrated into design & manufacturing systems and therefore provide an easy way of converting design features into manufacturing features [GDRN97]. As an example, Parametric Technology Corp. announced the machining feature based module called "Expert Machinist" as part of its commercial CAD/CAM system Pro/Engineer. Conceptually, users can assign detailed fabrication parameters to parametric design features and can therefore create G-codes easily.

The conclusion of the discussion on feature usage in design and manufacturing follows:

- If it is applied to a specific domain, manufacturing features are a good information media and can be achieved from the design model in a fairly automated way.
- The manufacturing feature model can be an attractive alternative for the electronic data exchange between design and fabrication when fabrication-specific information is needed to guarantee successful fabrication.

2.3 Feature Based Approach in the Machining Domain (Automated Process Planning, Quotation)

As discussed in the previous section, a form feature model in the machining domain can be generated in a fairly automated manner. Some implementations of feature-based approaches on the manufacturing side are now discussed, e.g. process planning for machining applications.

Computer-Aided Process Planning (CAPP) serves as a bridge between design and manufacturing [AZ89]. The process plan is used to evaluate manufacturability and compare several alternative designs.

---

[5] Parametric Technology Corp.'s Pro/ENGINEER was the first parametric modeler on the market. Most new and existing CAD tools (including SDRC, Unigraphics and CATIA, etc.) have been equipped with parametric tools.

[6] ACIS and Parasolid are the most popular solid geometric kernels. Based on these geometric kernels, a large number of CAD/CAM tools have been developed.

The selection and sequencing of operations necessary to transform the raw material into a finished part is its primary objective. Shah and Mäntylä [SHMA95-2] explain a process planning system framework based on planning method, depth, level and time scale, and discuss issues related to process planning features.

Despite a good deal of effort, there are presently no automated process-planning systems capable of automatically performing the complete planning task. The XCUT [HW90] system generates process plans for the production of machined parts from a feature-based part description. The input to this system will represent each part as a collection of manufacturing features (a feature model) where each feature is a region of the part (a form feature), which has some degree of manufacturing significance. XCUT provides operation planning for prismatic parts on multi-axis CNC milling machines. The design by feature approach is adopted with the solid modeling methodology in XCUT and production rules in the expert system (IF-THEN type) are used to generate operation planning [GRDN97].

In the First-Cut project, features generated in a feature based design system, are coupled with their corresponding process plans. The system creates a final part and process plan concurrently [CTM88]. The Next-Cut system [CTB92] includes a feature representation under the environment where multiple concurrent engineering activities are carried out based on the feature based design model, in addition to conventional process planning.

Kramer [KJ86] shows how manufacturing features can automate the downstream machining activities. He developed a program based on specific machining features, e.g. drill, groove, pocket, etc. The program, written in Lisp, treats the machining features as objects with assigned attributes. For example, a hole has attributes of xy location, diameter and depth. The system is quite simple to use and intuitive for anyone who has any machining experience. Machining features can be edited and some limited error checking exists. Once the user has defined the machining features, the last operations, which perform process planning and G-code generation are done automatically.

Fixture and Setup planning are also very important for successful machining. The fixture planning requires consideration of clamping devices and formulating the conditions that are needed to insure proper fixtures. Setup planning involves determining the various setups in which the part will be machined. Many researchers [Chang90, YM90] provide conditions for holding the workpiece with various clamping devices based on the intermediate workpiece geometry. These results can be coupled with solid modeling methodology and manufacturing features to produce reasonable fixture and setup planning.

Cost Estimation and process plan evaluation are also possible tasks in feature-based approaches. In the work of Gupta et al. [SMN94, DGN95], several alternative operation plans are evaluated by quantitative measures. Multiple feature models generated through feature recognition are needed to evaluate and select the best feature set. Features in the model are mapped to machining operations. Achievable machining tolerances and production cost and time are calculated and used to evaluate process plans for machined parts.

As seen in this section, the feature based approach has been aggressively pursued in a number of venues throughout the whole downstream fabrication-side process. The examples cited in this section show that the feature model in the machining domain could be used as input to automated manufacturability analysis.

2.4 Internet Based Design and Manufacturing

In this section, some distinctive internet based activities in design and manufacturing are described. There is no disputing that technologies related to the internet are being actively developed at this time. There are several interesting examples of ongoing work dedicated to the concept of using the web as a CAD/CAM development platform.

As an effort to integrate the design and manufacturing environments, the world's first web based CAD/CAM system is probably the "CyberCut" system [WW98] developed at Cal-Berkeley. The CyberCut web site (http://cybercut.berkeley.edu) provides a working system with CAD/CAM tools for web based design and manufacturing. Despite some functional deficiencies compared to commercial CAD/CAM systems, CyberCut provides a single working environment. Design is guided by several machining related rules (such as minimal corner radius). This system is an academic research tool, not a commercially viable system, but it still shows the potential of the concept.

Kim et al. [KKKKO98] developed a web-based collaboration system CYBERVIEW that uses the Virtual Reality Modeling Language (VRML) for sharing geometric information. Since the VRML format is one of the graphical and geometrical data formats that can be interactively viewed on the web using plug-ins, designers and engineers can collaborate with VRML models exchanged from the STEP format data in this system by adding markup on the graphical model or requesting information over the web.

The ARPA (Advanced Research Projects Agency) Manufacturing Automation and Design Engineering (MADE) program developed Internet-based tools, services, protocols and design methodologies that will allow contractors to compose teams of specialists from different locations and organizations as project needs arise. Based on the ARPA community member networks, the MADE program was tested to provide a collaborative environment of distributed facilities on the Web [CTG96].

2.5 Current E-Commerce Trends Related to the Metal Machining Industry

In spite of the high expectations of E-Commerce, most current E-Commerce sites only handle standardized goods. If a customer knows the product name, and attributes like size, color, and material, the customer is willing to buy the product. Products produced by job shops do not fall into this category and therefore present a relatively high barrier for entry into E-Commerce. Despite this difficulty, several E-Commerce sites are trying to run businesses targeted at job shops by providing the following types of services.

- Advertising: The user can find used or new machines advertised. Standard cutting tools and auxiliary devices for job shops are also sold. People, who want to sell or buy something linked to their job shop business, can post an advertisement on the web sites. The web sites may or may not charge fees for the advertising service. Hundreds of E-business sites provide this type of service. Imark.Com (WWW.iMark.com) is a good example for this type of business
- Search engine: Part buyers can get job shop contact information. Sometimes, buyers can get a specific list of job shops with a particular specialty. Most sites relating to job shops provide this service. At the National Tooling and Machining Association (www.ntma.org), the buyer can search the job shops according to the job shops' tooling and machining capabilities.
- Connecting part buyers and sellers: To date, only few web sites try to match the buyer and supplier of custom machined parts. MfgQuote.com (www.mfgQuote.com) is probably the only web site currently providing this kind of service. The main function of MfgQuote.com is to distribute buyer's RFQs

(Request for Quotation) to the registered job shops via the web. The business site can screen some job shops that are not capable of meeting the buyer's needs. This service can be regarded as an extended form of advertising. Job shops can contact the buyer, communicate with the designer via e-mail and make bids on the RFQ. Engineering drawings and e-mail are used as the primary communication tools. The process is somewhat time consuming and is wasted effort if the seller is not awarded the contract. It is the primary contention of this research that improved data descriptions are needed to enable truly successful E-Commerce in this domain.

2.6 High Level Information Interchange for Rapid Prototyping

Fabrication specific part data interchange has been tried in rapid prototyping for specific applications. Consider the 3D solid geometric information interchange for layered Solid Freeform Fabrication (SFF). As long as the geometry is valid for rapid prototyping, any geometric standard format can be used as the interchange format. However, since 3D Systems Corporation (www.3dsystems.com) adopted the STL format as the machine input language for the Stereo-Lithography Apparatus (SLA), the STL format has been used as a de facto standard of part geometry exchange for rapid prototyping in layered manufacturing. STL is a very low-level format and in that respect can be likened to G-codes. An STL file just contains unordered facets, which should construct a closed solid. There are several well known problems with STL: big file size, lack of connectivity, unaligned facet normal directions and lack of manufacturing related properties (material, tolerance).

To resolve the problems with STL, SIF_SFF (Solid Interchange Format for SFF) [SMS99] has been developed. SIF_SFF is a simple boundary and boolean operation based (CSG) representation of solid geometry. It can contain some properties like material, precision and structured geometry, such that the fabricator can create SFF processes without explicit or additional information. Basically, STL_SFF can substitute for STL and generate files that can be directly used by SFF machines.

For rapid prototyping in 3-Axis Milling, shapes of volumes to be removed are generated during process planning and used directly for NC-programming to generate actual machine codes. Transferring removal volume instead of final part geometry may make the fabrication processes smoother and easier.

Kramer defines MRSEVs (Material Removal Shape Element Volumes)[KRAMER92]. MRSEVs are volumetric features corresponding to machining operations on 3-axis milling machines. Kramer represents MRSEVs with EXPRESS (the official STEP information modeling language) and STEP form features. The STEP based format is adopted by Regli et al. [RGN94] and used for interchange of machining features extracted by feature recognition.

CyberCut can generate a geometric model by subtracting machining features from the workpiece. These procedural removals of machining volumes are saved in the SFF_DSG format. The SFF_DSG representation describes a model part with DSG (destructive solid model) methodology.

The data representations for data interchange mentioned in this section show that domain specific information (manufacturing process-centered information) has been used to enhance the interchange of required data and to make the subsequent data processing smoother and easier in manufacturing.

2.7 Standard Data Formats

Since, as seen in the case of the MOSIS service, having a standard format for data interchange plays an important role, currently used and successful standard formats in various areas will be reviewed and related to the goals of this research.

2.7.1 Page Description Language (Postscript, PDF, HTML)

An interesting analogy can be made between NC programming and page description languages. The most verbose, yet most exact, method for describing the layout of a page is Postscript. Postscript is a widely used standard for printers and usually guarantees fidelity between the exact layout of a page, as seen on the computer screen, to that which the printer produces. The widely used PDF format also belongs to this category. Another type of model is HTML. HTML is a markup language widely used on the World Wide Web. HTML usually guarantees that the intent of the sender will be satisfied, but leaves the details of how to achieve the results up to the receiver. Two different browsers will almost certainly not result in exactly the same look for the same HTML file. The advantage of HTML is that the file size is significantly less than in Postscript, a feature that makes it very desirable when bandwidth is limited. In terms of data handling, transferred data from design to manufacturing in the metal cutting industry should be similar to HTML. As long as the design intention is understood, the manufacturer can apply different detailed strategies for executing the plan for achieving the goal.

2.7.2 Product Data Standards (IGES, STEP, AP224, tolerances)

The data describing the part must be provided in a format that is accurate, complete and unambiguous. Furthermore, the usefulness of this information is proportional to the ease with which it can be read and interpreted in an automated manner by machines.

While engineering drawings in either paper or electronic format (for example, DXF format) do present a time-honored method of data exchange, they are often incomplete, inaccurate, ambiguous, and they are certainly not easily read and interpreted by machines.

The tremendous variety of software systems used in design and manufacturing dictate the usage of some method for electronic data interchange. Typically, design and manufacturing are done on separate systems, therefore requiring data translation from the native data representation of the CAD system into a standard "neutral" file. The neutral file must then be translated into the native format of the receiving CAM package.

A number of standards are used to transfer information from one system to another. Geometry of various types can be transferred via the Initial Graphics Exchange Specification (IGES) or by using DXF files, a standard created by Autodesk, Inc. Corporation. Solid information can be transferred using the SAT standard created as part of the ACIS system [SPTE], and the XMT standard from Parasolids [PARA] or via STEP files. However, none of these methods provide the desired level of information exchange required for manufacturing of discrete mechanical parts. The geometry information is, by itself, inadequate to define a process plan for an NC machining program.

Representing features and design information has been addressed by both academic and standardization communities. Standards groups have been evolving means for describing generic classes of features for the purposes of data exchange. STEP is the International Standard for the Exchange of Product Model Data being developed by the International Organization for Standardization (ISO). PDES (Product Data Exchange using STEP) represents the activity of corporate, government, and standards development entities in the United States in support of STEP. However, at present, the standard is still evolving and there is no definitive structure for representing and exchanging all the relevant information. A discussion of the STEP Form Feature model can be found in [SHMA95, SSSMW96]. The STEP AP224 standard [STEP_W] is also a possible approach since it is designed to allow process planning using machining features. It is a well-defined standard, but is not yet a generally accepted format. It is also rather challenging to implement.

As an effort of implementing the overall STEP standards in the NC machining domain, STEP-NC[7](STEP AP238, ISO14649) is designed to replace G-Codes (ISO6983) as the language of NC machines. Added to geometric information of the finished part and the raw material, STEP-NC data is composed of mutually sequenced tasks, which link to machining strategies and machining shapes (Machining Features). All information is described in terms of STEP related standards. This integrated data structure seems to contain sufficient information to directly run NC-machines equipped with an intelligent NC controller and are expected to replace G-codes with their verbose, inconsistent and "information-poor" style.

Although STEP AP224 has existed as a standard for a number of years, it has yet to be adopted by industry. So STEP-NC, which is based on the STEP AP224 and other newly implemented standards, may also need a long time to be practical enough to be accepted by industry.

Standard schemes have either not been employed to represent features or have not been mature enough; therefore, existing systems cannot be directly used by many commercially available CAM applications. Research on a feature-representing standard needs more attention from researchers.

2.7.3 WWW Standards (HTML, VRML, SGML, XML)

In this section XML [W3C2] is compared to other markup languages in terms of usability in software development and web compatibility. XML is the document-processing standard proposed by the World Wide Web Consortium (W3C) [W3C1].

XML and HTML are both derived from SGML (Standard Generalized Markup Language) and while HTML has permitted the rapid growth of the Internet by making documents universally "human readable", XML is designed to allow data to be formatted such that it is "machine readable". Unlike

---

[7] Visit STEP Tools, Inc. at www.steptools.com for details. STEP-NC is discussed more in Section 4.7.

HTML, which has a fixed set of tags that identify the characteristics of text; XML allows users to define their own tags, thereby making it possible to share databases via the web in a format that makes machine interpretation possible.

The format of the XML document can be defined as a Document Type Definition (DTD). Therefore, applying the DTD to the XML document can be used to check the syntactical correctness of the NCML file. Moreover, the DTD document can be stored at any place on the Internet and referred to by users worldwide. XML provides an agreed upon syntax and the DTD provides a definition for the structure and content of the particular data.

A number of applications are being developed by disparate groups of users, e.g. MathML for the definition of mathematical equations, SML for the steel industry, etc.

An XML document can be displayed in popular web browsers like Microsoft Internet Explorer (Version 5.0 or up) without any modifications and programming efforts. Moreover, there are many supporting tools to make XML documents available on the Internet. To date, at least two popular XML formatting languages for improving the appearance of the XML document on the web browser are available: CSS (Cascading Style Sheet) and XSL (eXtensible Style Language). For programmers who must develop applications to process the XML, APIs in the most common computer languages, such as C++ and Java are available for data parsing and storing the XML document in the database system [ECKS99, MORR00]. The Architecture of XML working environment is illustrated in Figure 2.2.

Viewing and handling 3D objects is possible using certain types of viewing applications for a specific format. The tendency to provide richer information on the World Wide Web has existed for different types of information like sound clips, images, movies etc. For a format to be used on the web, it should be platform-independent and computationally inexpensive.

VRML is a platform-independent file format for sharing 3D worlds on the Web. VRML can be a prime candidate for providing interactive visualization of 3D geometry on the web. VRML, introduced in 1995, is a language for describing geometric information in a hierarchical structure and providing interactive controls (displaying, interaction, executing programs) in its geometric contents. VRML is capable of creating virtual worlds and is hyper-linked with the WWW. VRML viewers[8] are companion applications to standard WWW browsers for navigating and visualizing.

While some geometric definitions[9] are not supported in VRML representation, VRML is one of the easiest standard ways to handle 3D geometry on the web at this moment. Many researchers [ARSNRT98, HGS98, KKKKO98] show how VRML has been used to build collaborative environments on part design and manufacturing by displaying 3D geometry on the web.

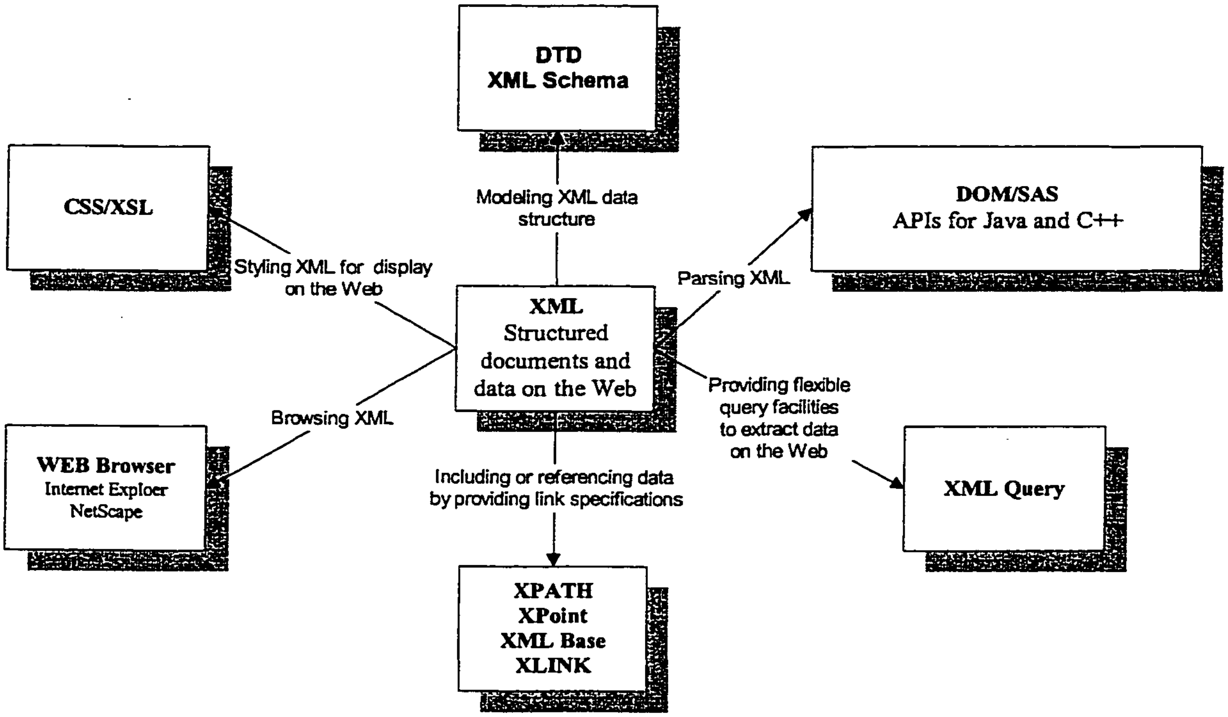


Figure 2.2 XML and its companion standards applicable in the Internet environment

---

[8] COSMO Player and WORLDVIEW are the most common VRML viewers used with common standard web browsers.

[9] Arc geometry must be defined as segmented lines in the VRML definition.

CHAPTER 3

SYSTEM ARCHITECTURE

This chapter discusses the future working environment in which NCML will be used and the requirements for successful implementation. The designer, who is often under intense time pressure, is eager to obtain prototype parts for the purpose of testing, as soon as possible. E-commerce methodology offers the possibility of accessing an expanded pool of suppliers with the potential for faster delivery at lower cost.

In the vision of this research, it is assumed that the fabricator has a computer attached to a CNC machine tool. The computer on the machine tool is also equipped with a database system that stores all machining conditions like feed rates and spindle speeds used for each type of material and cutting tool. The computer is connected to the Internet and the machinist can be simultaneously machining a part while searching the web for future jobs. From an E-Commerce site, the fabricators can find appropriate parts for their machine tool. Since the part information provided by the E-Commerce site is in the correct form to automate the machining process, the fabricator can use machining strategies stored in the database to quickly develop an efficient machining program.

The quality and format of the information delivered to the fabricator is central to the success of this vision. The information supplied must be complete and easily translated into machining commands. To work in an E-commerce environment, the format must be compatible with current web based technologies.

This chapter is devoted to describing the specifications and features of the new data format and software systems. Software systems are implemented to validate the usage of the new part data format and to show how the format can make the machined part fabrication process possible in an E-commerce environment.

This chapter begins with a discussion of the user requirements in the E-commerce environment. Then it explains how software systems and data representations will be constructed in response to these requirements. Possible working procedures with the developed software systems are also presented. The procedures, which lead to the achievement of the desired Clean Interface for custom machined parts are explained by showing data interactions among the software systems and related users. System specifications for the software developer are given at the end of this chapter.

3.1 System Requirements

In the assumed E-Commerce situations, the participants in the commerce of custom machined parts have at least two roles - designer and fabricator. The part designer can be a buyer and the part fabricator can be a seller of machining resources and labor. The requirements of each of these two players are now discussed.

3.1.1 Designer and Fabricator Needs

The part designer is always interested in getting parts made as quickly and as inexpensively as possible. The fabricator needs to turn design information into physical parts by making efficient use of manufacturing resources and labor.

Simulating manufacturing operations in the design phase can be a way to check feasibility of part fabrication, a capability that meshes nicely with concurrent engineering. Designers are responsible for creating a design that fabricators can produce. Sadly, this is often not the case and designers are often guilty of producing designs that either can't be produced or are unnecessarily expensive to make.

The fabricator should not have to rework the design in order to produce it. Rather, the tasks on the fabrication side should rather be "post processing of delivered information from the designer", such that a skillful machinist can focus on improving fabrication efficiency rather than solving design related problems or recreating information that should be readily available.

Process efficiency issues are important to both buyers and sellers of machined parts in either electronic or conventional business. Since the most important thing for the designer is to receive the parts in a timely fashion, the designer may want to provide part information in the format that is most useful to the fabricator.

To be competitive, sellers must be able to efficiently execute their business related activities. This task related efficiency is directly related to their information handling capability. The key to a successful fabrication business is the ability to easily handle the information delivered by the buyer.

The needs of the designer and the fabricator are summarized as follows. The designer should govern the macro aspects of fabrication and the fabricator should only have to focus on the micro aspects of fabrication without any confusion in understanding the design intent. To achieve this, design information should be tested appropriately before being sent to the manufacturing side and more fabrication specific information may need to be added to the conventional part information delivered to the fabricator. The information should be delivered in a format so that the fabricator can use it efficiently.

3.1.2 Requirements for the Part Interchange Format

The part information delivered from the designer to the fabricator should contain unambiguous information for both the designer and the fabricator. The following characteristics are required in the new data interchange format for the machined part.

- Mutually Understandable

As seen in the process proposed in Figure 1.2, the new data interchange format is the core element of the clean interface. Compared to the conventional process, the main difference is that the new process uses the data interchange format as a bi-directional communication tool. The new process should be more compressed, removing some serial processes in the middle. The data interchange method should be understandable to both the design and the manufacturing sides and also serve as a direct input for the necessary operations.

- More Fabrication-Centered Information

It is a basic requirement that the data representation being proposed should be able to convey enough information to fabricate the specific machined part. More than geometric information is required to fabricate a machined part; for instance, GD & T (Geometric Dimension and Tolerance) and surface finishing.

Based on the role separations, where the designer governs the macro aspects of fabrication and the fabricator governs the micro aspects, what kind of information must be delivered to the fabricator? If more information about the part is provided to the fabricator, it is easier for the fabricator to develop the part fabrication process. However, a compromise must be struck; generating more information in the design phase places a greater burden on the designers. On the other hand if there is insufficient information provided to the fabricator, it increases the possibility of inappropriate fabrication. For example, information about jigs and fixtures is essential for process planning, but these are not required in the design phase. This type of information is too fabrication oriented to be added to the part model by the designer.

- Need for a Standard Data Format

As long as the fabrication-side activities are performed smoothly using the part data interchange, the format of the part data interchange does not matter. However, there are several reasons for using a standard format. First of all, we only need to focus on the data contents, not the style or information format. Second, by using a standard format we may avail ourselves of the use of already developed utilities, software systems, or APIs for parsing data, viewing documents, etc. Furthermore, considering the Web based information transactions, the standard format being chosen would be better if it is compatible with the current web based technologies.

3.1.3 E-Commerce for Fabricating Parts

The design and fabrication will be coupled with E-Commerce between the part designer (buyer) and the fabricators (sellers). The E-business environment provides an ideal situation for functionally separating the fabricator from the designer.

3.1.4 Implementation Scope-Machining Operations and Part Shapes

The core requirement of this research is to develop a data format capable of conveying enough information to not only describe the design, but also define enough of the manufacturing process to avoid burdensome communication that causes an increase in either cost or delivery time. However, a series of different manufacturing operations (e.g., milling, drilling, lathing, EDM) may be needed to make a mechanical part. To deliver all the required information about any mechanical part for manufacturing may be difficult. Instead of considering all manufacturing processes, we focus on the most commonly used metal cutting process - milling and drilling. Specifically, the parts considered in this study are limited to those manufactured with a standard 3-axis CNC milling machine. Part shapes are limited to the prismatic part domain (2½-D machining); sculptured surfaces are not currently included.

3.2 System Design-Structural View

The developed system architecture is presented in this section by explaining each component of the system and their mutual relationships.

3.2.1 System Architecture

The system pursued in this dissertation is implemented by a new part data interchange format and three software systems as seen in Figure 3.1.

As a core component, a new part data interchange format, called NCML (Numerically Controlled Markup Language) is proposed to ease electronic part data interchange and provide better communication between designers and fabricators of machined parts.

If this data interchange approach is able to convey the part information in both directions, the "Clean Interface" can be achieved by a functional separation between design and fabrication. This distinction of functions can be embodied into each software system that represents activities of the associated roles - the designer and the fabricator. These design- and fabrication-side software systems are named FACILE[1]/Design and FACILE/Fabricate respectively. FACILE is used to name these two systems together.

FACILE/Design is a part design software system that performs design related tasks like geometry creation and editing, and exports an NCML file of the resulting part. FACILE/Fabricate is a software system that imports an NCML file to be used by a fabricator to create machine-operating codes (e.g., G-codes).

---

[1] FACILE is the acronym for "A Fast Associative Clean Interface Langue and Environment" for Discrete Prototype Fabrication. FACILE refers to not only the proposed environment for part fabrication but also to the name of the CAD/CAM systems developed for validating the proposed environment in this dissertation. The previous work regarding the FACILE system has been done at the Design and Manufacturing Lab at the University of New Hampshire[JERARD98, JERY2000, RYJE2001].

To introduce E-Commerce to the suggested part design/fabrication process, an E-Commerce model for machined parts - a web based computer application - is needed to demonstrate the role of the E-merchant who connects the buyer to the seller. This is a "business to business" (B2B) application. This web-based application is indicated as E-Mill[2] in Figure 3.1. The E-Mill service provides concrete communication channels between designers and fabricators, based on the mutual understanding of part information in the NCML file. At the beginning of this research in 1997 it seemed necessary to create a B2B application to prove feasibility. Since that time, many B2B providers have been started (and some subsequently stopped), with at least one focused on connecting buyers and sellers of custom machined parts (www.mfgquote.com).

Therefore, the feasibility of the concept is tested with three prototype software systems: FACILE/Design, FACILE/Fabricate and E-Mill. Figure 3.1 shows the relationships and the roles of the components.

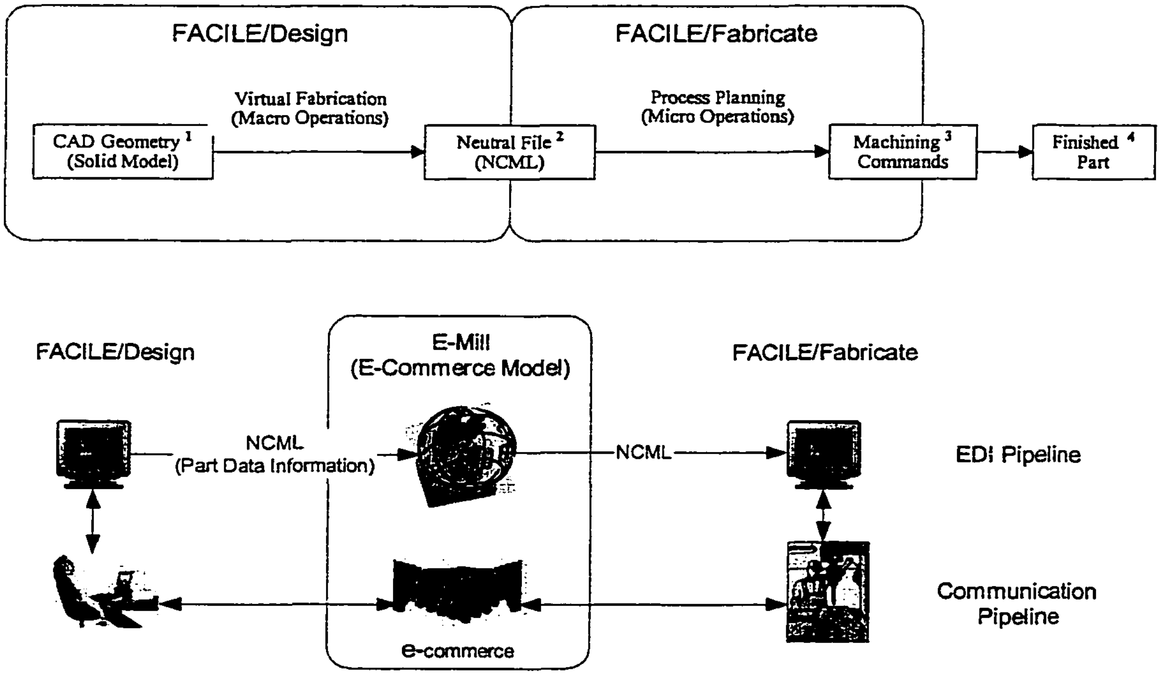


Figure 3.1 Overall system architecture

---

[2] E-Mill is the name of the E-Commerce model and implementations for the machined part fabrication.

3.2.2 Basic properties of NCML

In addition to the part geometry, the following items need to be included in NCML

- Part Specifications –Information on part material, GD &T, Designer, etc.
- Process Planning- a simplified process plan, order of the machining operations.

The data in NCML must provide user-friendly information to the fabricator, help automate downstream activities, and make data handling efficient. By adopting Object Oriented Programming (OOP) techniques, information in NCML is conceptualized as objects for the machining/manufacturing of features. Moreover, these machining features are organized in a hierarchical structure. The hierarchical structure makes it easier to transfer data components grouped or related in the parent-child data. How the intended data is encoded in NCML is explained as follows:

- Manufacturing Features: Manufacturing features are used as a unit of data grouping. These include not only machining features like pockets, holes, contours, etc but also other part–specific information like tolerances, work piece, cutting tools, etc.
- Hierarchical Structure: The machining features in NCML are structured hierarchically. These machining features are supposed to be machined out from an intermediate work piece according to the machining order in the structure.

As suggested earlier, using a well-defined document standard has advantages. The XML format is used to define and to structure NCML.

- XML: XML is extensible. New components can be easily added to NCML. Since the current implementation of NCML is quite simple, the extensibility of document contents must be considered for future practical application development.

A detailed explanation of the features of NCML is deferred until Chapter 4.

3.2.3 FACILE/Design Approach

FACILE/Design must include a number of functions relevant to the designer activities in Figure 3.1. The functions include: creation, edit, and simulation. Of course, NCML export from the created part model in FACILE/Design is essential. As a design software system, FACILE/Design should provide a graphical user interface.

The "Design by Features" methodology is adopted to achieve the feature information in NCML. The inclusion of a solid modeler in FACILE/Design provides powerful feature modeling and simulation capability. Machining features can be modeled by solid geometry. Therefore, the use of solid modeling capabilities make it easier to define the machining feature.

The solid operations also provide an effective validating tool for machining operations. By subtracting each machining feature from the intermediate part solid geometry according to the operation orders in the NCML structure, machining simulation can be done in a "Macro" sense. While individual tool paths are not simulated, manufacturing features can be created as a single solid, which can be subtracted from the raw material block geometry.

3.2.4 FACILE/Fabricate Approach

The fabricator must post-process the macro part definition in NCML into a micro toolpath program, simulate the resulting tool paths to check for errors, and estimate the machining cost. Each of these functions should, ideally, be done with a minimum of human intervention.

The output of the post-processing is G codes, the language understood by most NC machines. The competitive advantage of the FACILE/Fabricate process depends on how accurately and efficiently the Macro program given in a NCML file is processed into the Micro program. Therefore, the fabricator should be able to edit the Macro process plan in NCML to optimize the machining process order, and to apply customized experience-based fabricator-specific machining conditions. The individual capabilities of the fabricator will be accommodated by the adjustment of attributes associated with detailed process planning. Each machining condition; spindle speed, feed rate, and cutting depth should be provided in a convenient way in FACILE/Fabricate.

Simulation at the micro level means that each individual toolpath must be processed. This toolpath simulation is used to verify the generated toolpath and avoid tool interference. Many researchers already have successfully demonstrated simulation [JHDS89, CJ98, CPSC98], and many commercial cutting simulation/verification software systems are available. Therefore, G-codes generated in FACILE/Fabricate are sufficient to prove feasibility, and a simulation capability is not currently included.

The total time to fabricate a part includes set-up time, machining time and any additional time spent on de-burring, loading/unloading and etc. The fabricator should be able to estimate cost fairly accurately based on the foregoing time analysis plus material costs. This function may help the fabricator prepare accurate bids.

3.2.5 Communication Channel & E-Commerce

E-Mill is an E-Commerce model for a custom machined part whereby buyers can post RFQs and sellers can bid on jobs. E-Commerce here is based on the assumption that NCML conveys sufficient and required part information. In addition, a converter that turns NCML into a solid model of the machined part would greatly aid in helping the fabricator prepare a bid.

A matchmaking mechanism is also essential. Both the designer's need for quick fabrication and the fabricator's resource availability affect the business activity. Cost and delivery time are equally important in choosing a supplier. The matchmaking method adopted in E-Mill is based on an auction/bid platform similar to those of many successful e-Commerce sites.

While the goal of the Clean Interface concept in NCML is to require little to no dialogue between the designer and fabricator, as a practical matter some dialogue will be required. As seen in the MOSIS process, two-way communications between the designer and fabricators should be included in the E-Mill model to provide clarification. Email can certainly be used but the E-Mill site must have a mechanism for posting clarifications that can be seen by all potential bidders.

3.2.6 Summary of Functional requirements

FACILE/Design, FACILE/Fabricate and E-Mill are intended to support the processes of the designer, the fabricator and the e-market respectively for the proposed process based on NCML.

Figure 3.2 illustrates the basic ideas of the planned software architecture and communication based on the e-Commerce environment. The proposed process in Figure 3.1 can be mapped into the data flows and functions in Figure 3.2. Essential functions and possible communication modes are marked in Figure 3.2, according to the active roles in the process. Table 3.1 summarizes the essential functions needed implementing in the respective software systems.

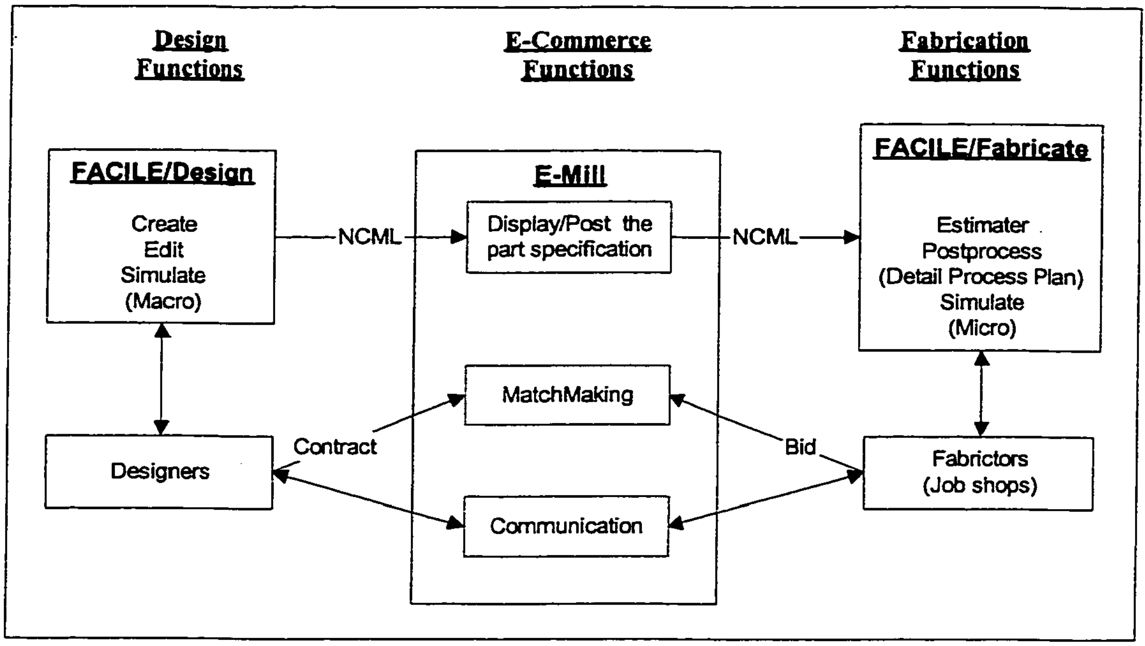


Figure 3.2 The NCML based E-Commerce model for the machined part

Table 3.1 Overall functional requirements

Applications	Functions	Descriptions
Facile/ Design	Create/Edit	Create/edit design part information and create NCML files
	Simulate-Macro	Simulate design intents in the macro view of the fabrication by executing solid DSG operations
Facile/ Fabricate	Quote Estimation	Estimate quote for the part described in the NCML file based on the each fabricator's capability The cost estimation tool can be supportive in the competitive business environment
	Postprocessor	Process the NCML to create real machine executable codes by applying process planning and applying actual machining conditions. G code generation is a part of this function
	Simulate-Micro	Simulate NC executable codes before machine operations
E-Mill	Post/Display the NCML	Display NCML files in the web-compatible format on the Internet. This may include graphical representations
	Matchmaking	Efficient matchmaking is a fundamental element in e-business.
	Communicate	To resolve design rule violations or queries between the designer and the fabricator, the communication channel is strongly recommended

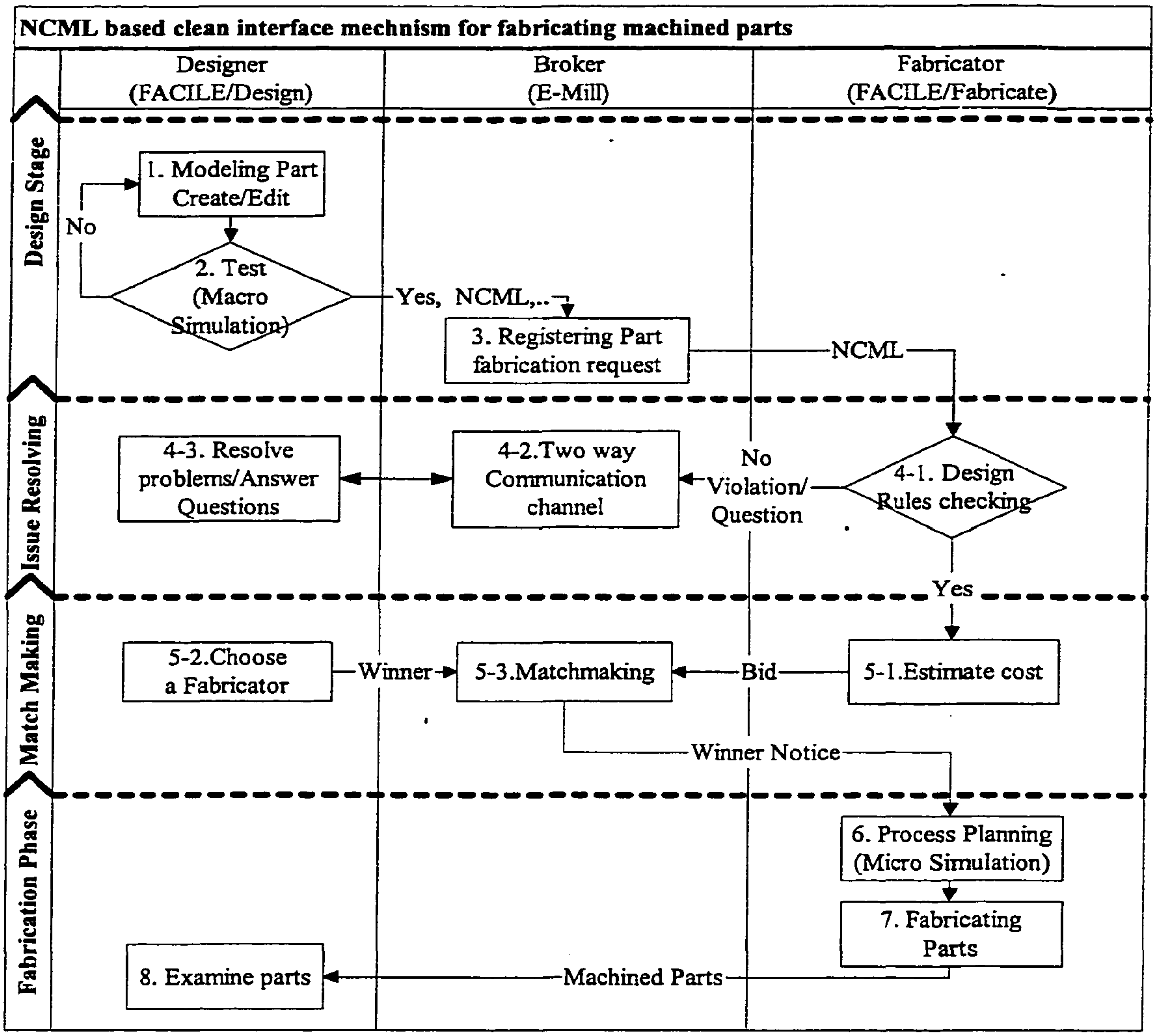


Figure 3.3 The proposed clean interface mechnism for fabricating custom machined parts

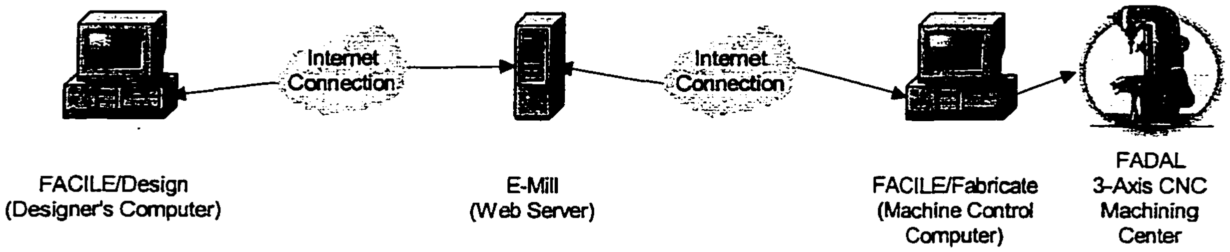
3.3 Clean Interface Mechanism

At least five components: two human components (the designer and the fabricator) and three software components (FACILE/Design, FACILE/Fabricate, and E-Mill) are related to each other in order to fabricate parts through the proposed process. User interactions with the software systems and communications between parties concerned are explained in detail in this section. A process chart in Figure 3.3, illustrates detailed tasks and communications. For simplifying purposes, the roles of the designer and

FACILE/Design (or Fabricator and FACILE/Fabricate) are put into the same component and named "FACILE/Design (FACILE/Fabricate)".

In Figure 3.3, a single process or task is contained in the box shape and the information carried between tasks is noted on the arrowed line. The horizontal direction flow shows how each software component is related to the others at each part fabrication stage and the vertical columns show which activities are required for each role in the proposed process.

3.4 Implementation Environments & Validation Architecture



	FACILE (Design & Fabricate)	E-Mill
Platform	x86 Family personal computer	
Operation System	Microsoft Window95, 98,NT, 2000	Red Hat 6.1(LINUX family) Apache HTTP Server 1.3.X
Programming Language	Visual C++ 6.0	HTML4, PHP3, Java
RDBMS	Not Relevant	MySQL 3.2.2
Common components	VRML2, XML	
Etc	Solid Modeler: Silver Engine 6.86 Xerces C++ XML parser(XML4C)	Cosmo Player Java2, JDBC

Figure 3.4 Software development environments

As seen in Figure 3.4, the hardware configuration of the proposed system is fairly simple. A web server is required to implement an E-Commerce model E-Mill. A computer with a Linux family operating system can be easily used to host a web server. In the E-Mill implementation, the Apache Web server is set up on a Linux operating x-86 family personal computer. A relational database system (MySQL 3.2.2) is used to keep data and to keep track of each transaction made on E-Mill. Document contents on each E-Mill HTML page are dynamically generated by the server side dynamic HTML script language –PHP3. PHP3 codes interact with the RDBMS, retrieve relevant information and create HTML documents. To enhance the fabricator's understanding, 3D geometry of NCML in the VRML format is adopted in E-Mill. Cosmo player is hooked on the E-Mill and is used to display the VRML file, which is used for visualizing NCML contents in 3D space.

The FACILE system only requires an Internet connected computer with Microsoft Windows. Visual C++ 6.0 is mainly used to build the applications. Internal data structures fully uses OOP features to utilize all its benefits. Publicly available XML document parsing APIs are used to export an NCML file from FACILE/Design and to import it to FACILE/Fabricate. In this implementation, the Xerces C++ XML parser (XML4C) is actively used to handle the NCML in FACILE systems.

Since solid modeling is essential to process several tasks like macro simulation in FACILE, the solid modeling geometry kernel, Silver Engine from Schroff Development Corp. is chosen to provide geometric handling capabilities.

To fabricate real parts for validating the FACILE process, the FADAL CNC machining center with the MDSI OAC is used. This machine is assumed to be attached to the FACILE/Fabricate computer.

CHAPTER 4

DATA INTERCHANGE FORMAT-NCML

NCML is based on XML, the document-processing standard proposed by the World Wide Web Consortium (W3C)[W3C1]. NCML has been devised as a web based data exchange format for custom machined parts. Since NCML is the core of this research and a key element for the clean data interface between design and manufacturing, this chapter is devoted to giving an explanation of the NCML Document Type Definition (DTD) and features in the NCML design.

For detailed information, refer to the following Appendices:

- Appendix A - The NCML DTD document.
- Appendix B – NCML reference manual. This part includes the NCML element tree, a sample NCML document of a simple machined part and descriptions of NCML elements.
- Appendix E – More NCML samples and machined parts fabricated from the NCML samples.

4.1 An XML Document -NCML

By using the XML format for developing a new data interchange format the following benefits are gained:

- XML is a well-constructed and rich data format: XML was specifically devised for the purpose of representing a variety of data contents on the web and provides a flexible framework for communication of almost any type of electronic data via XML documents [W3C2]. Creating a completely new protocol would be a redundant task, since much of the needed information concerning syntax and organization already exists in XML.
- An Internet compatible data format: XML lends itself to working with other specifications and tools. As seen in Figure 2.2, many web-based standards and programming APIs are equipped with XML to

provide usable web-based applications. Implementing an XML application on the web is easier with the help of these supporting technologies.

- Abundant programming resources and application examples: Although XML is fairly new and is still developing, many active examples and code samples are on the web. One of the most valuable sources for XML is W3C (http://www.w3.org/)[W3C1].

4.2 Structural View of NCML (DTD for NCML)

An XML document stores data in a hierarchical structure. The node on each level of the tree structure is called an *element,* which delimits its contents with a pair of *tags*[1]. DTD is used to build the data structure and contents in XML. Figure 4.1 illustrates a simplified structure of NCML. Figure 4.2 shows the detailed NCML architecture, using the Object Oriented Modeling notation of the Unified Modeling Language (UML), and models each XML tag as a class. Although the object diagram in Figure 4.2 is not enough to explain all NCML architectural features, such as, the sequence of children elements (tags) inside the parent element, it shows complete parent-child relationships of NCML elements.

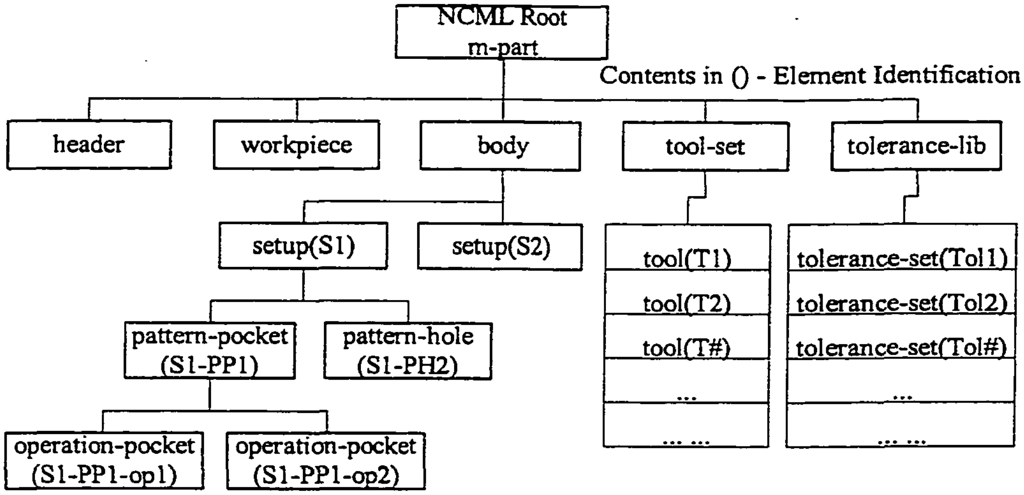


Figure 4.1 Simplified NCML structure and components

---

[1] Tags are the most obvious component in XML and are used to describe elements. Tags are usually used in pair. However for the empty element, a single tag can be used with a forward slash (/) in addition to the closing character (>).

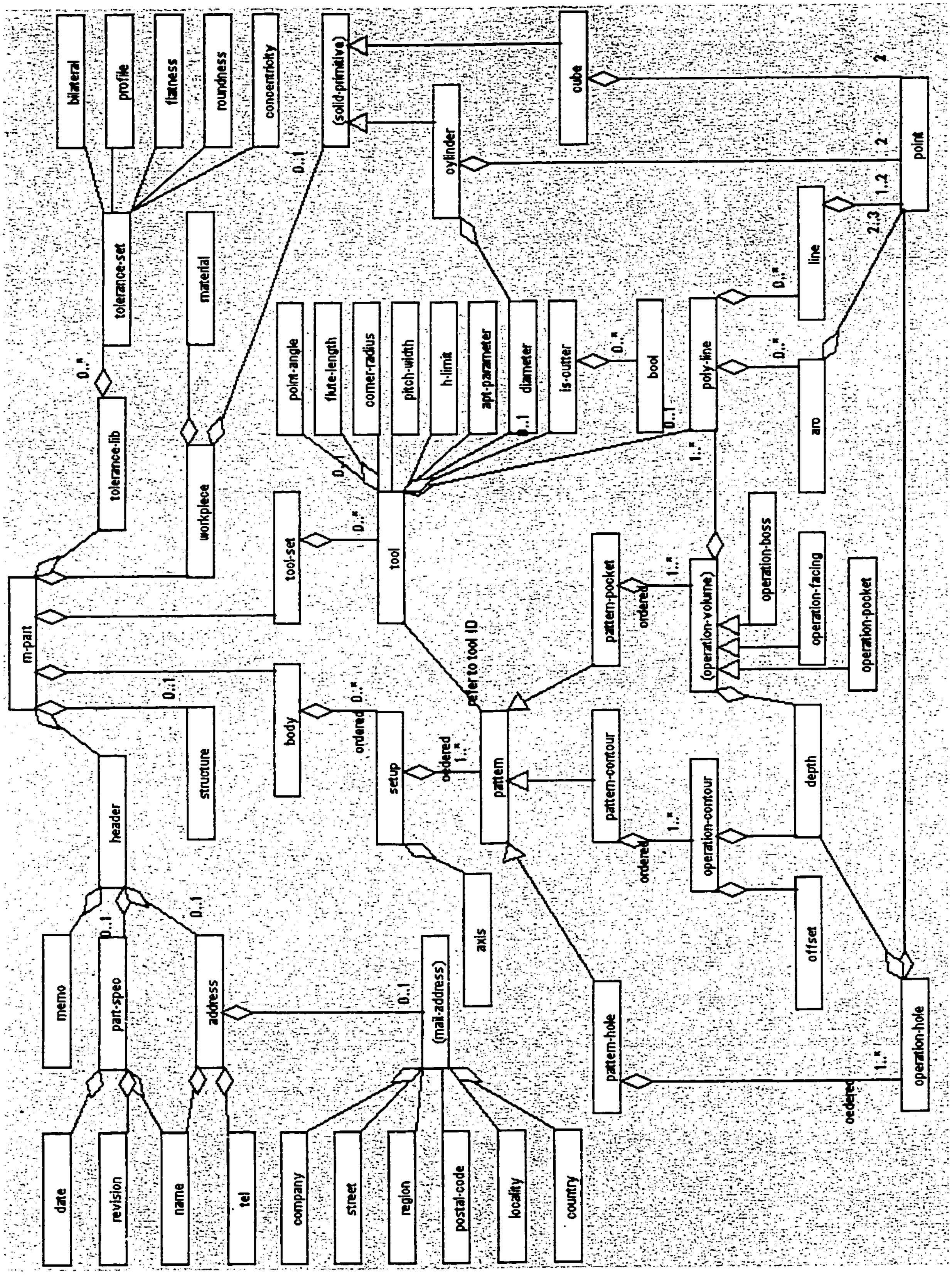


Figure 4.2 NCML element diagram

An NCML document is comprised of all information necessary to define a whole part. The root element of the NCML file is an *m-part*[2] which contains five top level elements, as seen in Figure 4.1 and Figure 4.2:

HEADER. The *header* element consists of text-type information about the part and its designer or buyer. It is composed of three elements - *part-spec, address,* and *memo*. The *part-spec* element includes information about part name, part number and release date (design). Contact information including an e-mail address is found in the *address* element. The memo element can be used to add additional information or an explanation of the part.

WORKPIECE. The shape and material of the work piece are defined in the *workpiece* element. First, the *workpiece* element specifies the shape of the work piece such as "cube" or "cylinder". Then real dimensions are followed in accordance with the type of work piece shape. The material type is also defined in this node. To adopt a more complex shape, such as, the shape of a molded part, importing external geometrical definitions such as IGES or STEP is needed but not currently implemented. The geometric definition of the work piece shape is described in Section 4.4, Geometry representation in NCML.

BODY. The body element is designed to describe how the work piece would be machined. Therefore, as seen in Figure 4.1, the body element is arranged in an intuitive format consistent with the language of machinists. From the top level, the body structure is broken down into "setup", "pattern", and "operation". The elements at each level are arranged according to a machining process plan. The *setup* and *pattern* elements are designed to contain manufacturing features in the *operation* element. These features are made more general by the *tool* elements under the *tool-set* element.

TOOL-SET. Commonly used cutting tools like ball-end and flat-end cutters can be defined in a *tool* element, as well as less commonly used tools, e.g., a dovetail cutter. The *tool-set* element is the container of *tool* elements. Each *pattern* element in the hierarchical structure of the *body* element refers to a *tool* element for the tool shape, which is used to define the manufacturing features with

---

[2] From now on, the NCML element's names are written in italic to differentiate it from other terminologies.

the geometric information defined in *operation* elements. Definable manufacturing features related to the *tool* element are discussed in Section 4.5, Manufacturing features in NCML.

TOLERANCE-LIB: In addition to geometric dimensions describing the feature, tolerance information is necessary before a part can be machined. A set of tolerances that can be applied to the geometry elements (*body, setup, pattern, operation*) are defined in a *tolerance-set* element. Several *tolerance-set* elements can be included in the *tolerance-lib* element. In each *tolerance-set* element, several tolerance values for surface tolerance, sizing tolerance, and concentricity tolerance, etc, can be added. Each structural element, *body, setup, pattern* or *operation*, has a reference to a *tolerance-set* element in the *tolerance-lib*.

4.3 Features in NCML

This section describes some of the design concepts embodied in NCML.

- Feature-Based Design and Data Exchange

NCML describes the physical object in terms of manufacturing features. Manufacturing features have been shown to be a convenient representation for manufacturing process planning [CHANG90, KN92, SHMA95-2] and for accomplishing related tasks such as NC code generation [FS20, GAIN99], setup (fixturing) analysis [CHANG90, GDRN97] and cost estimation [RGN94, GRN95]. With this approach an object can be generated by a CAD system that uses feature generation such as Interactive feature Identification[3], or Design by Manufacturing Feature[4]. Alternatively, an existing design can be converted to manufacturing features by automatic feature recognition[5].

---

[3] Picking geometric elements of a geometric model displayed on computer screen identifies features.

[4] The part design is achieved by manufacturing features. This approach forces the designer to think in terms of manufacturing operations.

[5] Feature recognition method extracts features directly from a geometric model. The problem is that little information from the design stage can be transferred to manufacturing. For example, dimensioning and tolerances are lost. The benefit is that a complete separation of concern between design and manufacturing is achieved. [SHMA95-1]

- Virtual Tool

Gaines [GAIN99] introduced a tool-centric approach to feature recognition. He extracts features based on cutting tool shapes in his system CUSTOM-CUT [GAHA99]. This system is equipped with a cutting tool library and machining features that can be machined by specific cutting tools in the library. NCML also uses cutting tool geometry to define manufacturing features. The concept that associates the manufacturing feature with a tool shape is called a "virtual tool". The virtual tool concept not only allows the user to easily and simply define machining features such as counter-bored holes, edge rounds, or tapered sides on a pocket or contour, but also simplifies the machining feature geometry and reduces the required number of manufacturing features (called *operations* in NCML) without losing the ability to express a diverse set of manufacturing features. The cutting tools actually used by the machine shop may or may not be the same as the "virtual" tool defined in the NCML file. Actual tool usage depends on the availability of each tool in the fabricator's tool crib. The feature definition along with the tool definition in the NCML file may be translated into several simple machining volumes in the machining process.

- Hierarchical Structure Definition and Process Planning

The *body* element is comprised of one or more *setup* elements. The *setup* element corresponds to an orientation of the work piece on the NC milling machine. In fact, the Local Coordinate System (LCS) is defined at the setup element. The LCS is an orthogonal right hand coordinate system defined by an origin in World Coordinate System (WCS) or Global Coordinate System (GCS). Any geometry declaration inside the *setup* element is defined relative to this LCS.

In each setup, machining operations can be grouped into a pattern. The *pattern* element is a group of similar *operation* elements that use the same cutting tool and share the same machining strategies. The pattern element is directly associated with a *tool* element defined in *tool-set*.

The *operation* elements are the fundamental machining features and can be one of three basic types: hole, contour, and pocket. These three fundamental operations can be used to create holes, pockets, bosses, slots, facing, and side milling operations. The pocket is used not only to represent what is usually called a pocket, but also to represent a large variety of milled shapes such as steps, profiles, slabs, etc.

The hierarchical structure and the order of elements under the parent element define a plan for fabricating the part; including setup preparation, tool changes and the order of machining operations. This means that NCML embodies not only part geometry and manufacturing features but also the basic machining process plan.

- Geometric Dimensioning and Tolerancing (GD&T)

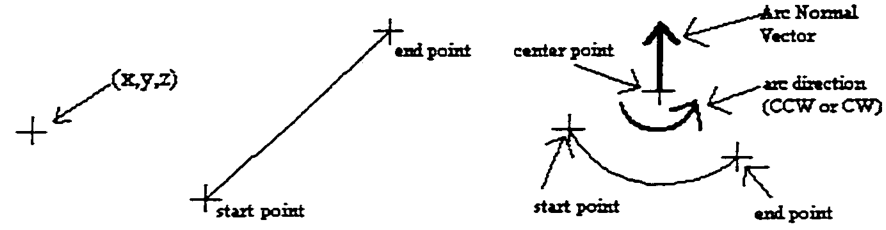
NCML is developed for the purpose of delivering data for manufacturing parts to the fabricator with minimal contact between the buyer and the seller. The inclusion of tolerance information is necessary since tolerances are related to cost, machining time, etc. NCML includes the tolerance information inside the *tolerance-lib* element and this information can be used in cost estimation and in planning the machining operations. In the E-commerce business environment, being able to develop a fast and accurate quote is essential and tolerance requirements can dramatically affect the quoting process. Fabrication time and associated costs are greatly increased as allowable tolerances are decreased.

- Other factors

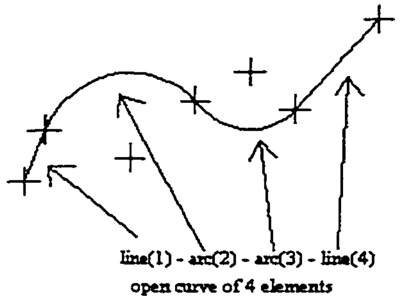
Since an XML file is in simple readable text format, NCML can be read and edited with any commonly available text editor. The extensibility of XML is applicable to NCML. The information which is usually transferred in the non-electronic formats like numerical tables, fabrication specifications will be easily included in the NCML definition.

4.4 Geometric Definitions in NCML

Any NCML element, which includes shapes or geometric references, is defined with the aid of *point, line, arc, and curve* elements. A *point* can be interpreted as a scalar or a vector quantity in the specified coordinate system defined in space. A *curve* is comprised of lines and arcs and may be either open or closed. For example, a hole is defined by a point, a pocket by a closed curve, and a contour by either an open or closed curve. See Figure 4.3 for the basic geometric elements in NCML.

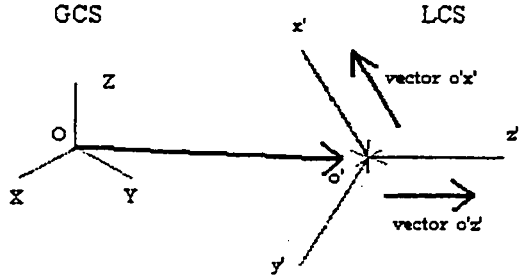


(a) point element (b) line element (c) arc element

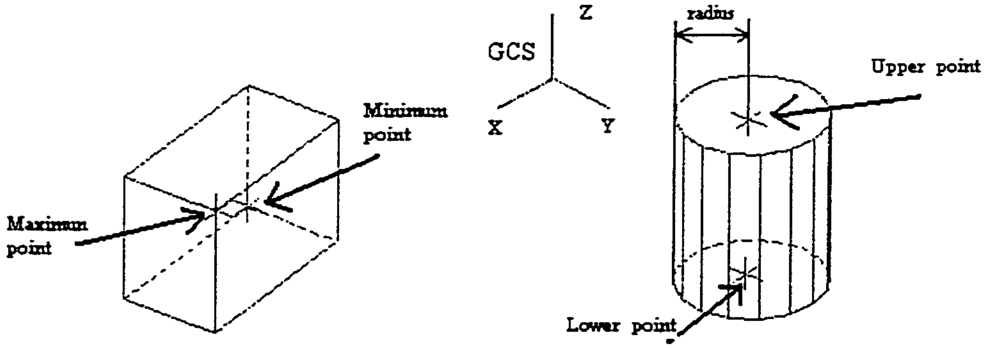


(d) curve element

Figure 4.3 NCML basic geometric elements



(a) axis element - three point elements
o' - LCS base point
vector o'z' - z-axis direction
vector o'x' - x-axis direction of LCS
o',x' and z' are defined in GCS



(b) cube element
- two diagonal point elements

(c) cylinder element - two axial point elements and radius

Figure 4.4 Axes, cube and cylinder elements

An *axis* element is created to define an LCS, which is an orthogonal right hand coordinate system and composed of three *point* elements (two of these point elements are used as vectors) in Figure 4.4-(a). One *point* element is used to place the origin point of a coordinate system and two vectors indicate z-axis and x-axis direction each. The *axis* element is used in the *setup* element to set an orientation of the work piece for each NC milling machining setup.

The cube and cylinder elements define simple solid shapes with point elements and positive real number attributes (See Figure 4.4-(b)), which are used to define typical work piece shapes in the GCS.

4.5 Manufacturing Features in NCML

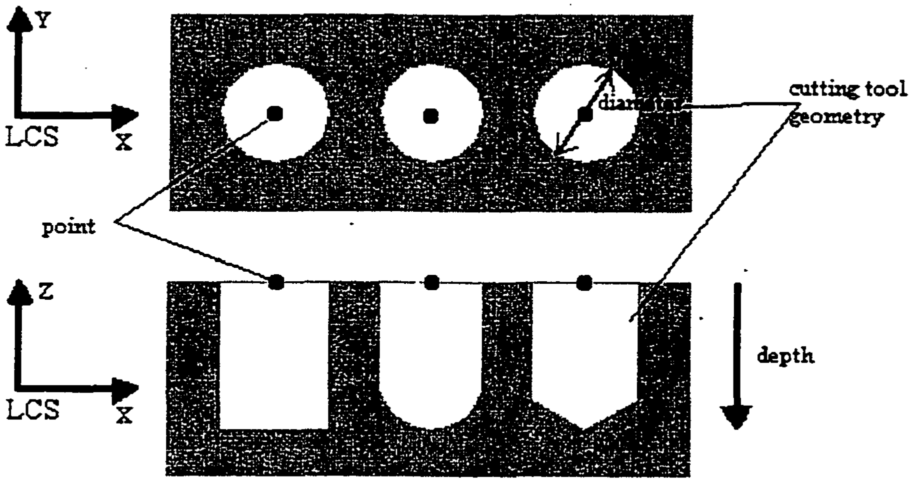
The manufacturing feature and its associated cutting tool define an operation in NCML. The manufacturing features (e.g., Hole, Contour, Pocket, Boss and Facing) defined in the *operation* element are currently focused on 2-½ machining. All operations are associated with geometry consisting of points or curves.

NCML has five operation types: hole, contour, pocket, boss and facing (two of these, boss and facing are actually derivatives of the pocket type). Figures 4.5-4.9 illustrate each operation type. These figures include a list of the attributes which were modeled as lower level NCML elements or generic XML attributes which are attached to the element. The top and front orthographic views of an example of operation elements are also shown in the figures. The LCS shown is inherited from the setup element according to the NCML hierarchical structure, so that every element which includes geometry is defined based on the LCS. The manufacturing feature volume is a function of the geometry in the NCML operation and the cutting tool shape is defined by the "virtual tool". This tool shape is associated with the *pattern* element and is inherited by the *operation* elements belonging to the *pattern* element.

Figures 4.5-4.9 show the manufacturing features as though they were voids in a block of material. However, NCML operations are defined as closed volumes, which are to be Boolean-subtracted from the work piece.

- Hole Feature (*operation-hole*).

The hole feature represents a cylindrical solid shape which is to be machined from a block of material by a drilling operation or consecutive drilling-like operations. The hole location point and depth are the only attributes needed in the *operation-hole* element to determine the feature solid shape. By applying different tool geometry, this simple *operation-hole* element can express quite different shapes. For instance, a hole with a counter sink, a counter bore, or steps. Figure 4.5 shows examples of holes with the same *operation-hole* element definition but with three different cutting tool geometries. The actual cutting tools that would be commonly used to create the holes include drills, center drills, counter sinks, taps, reamers and boring tools. The choice of tool is correctly left to the discretion of the fabricator, based on the hole size, virtual tool shape and required tolerances. The preferences and capabilities of the individual job shop will also play a role in creating a detailed manufacturing plan. This is consistent with the underlying philosophy of NCML; i.e. defining the finished product, but leaving the implementation details up to the fabricator.



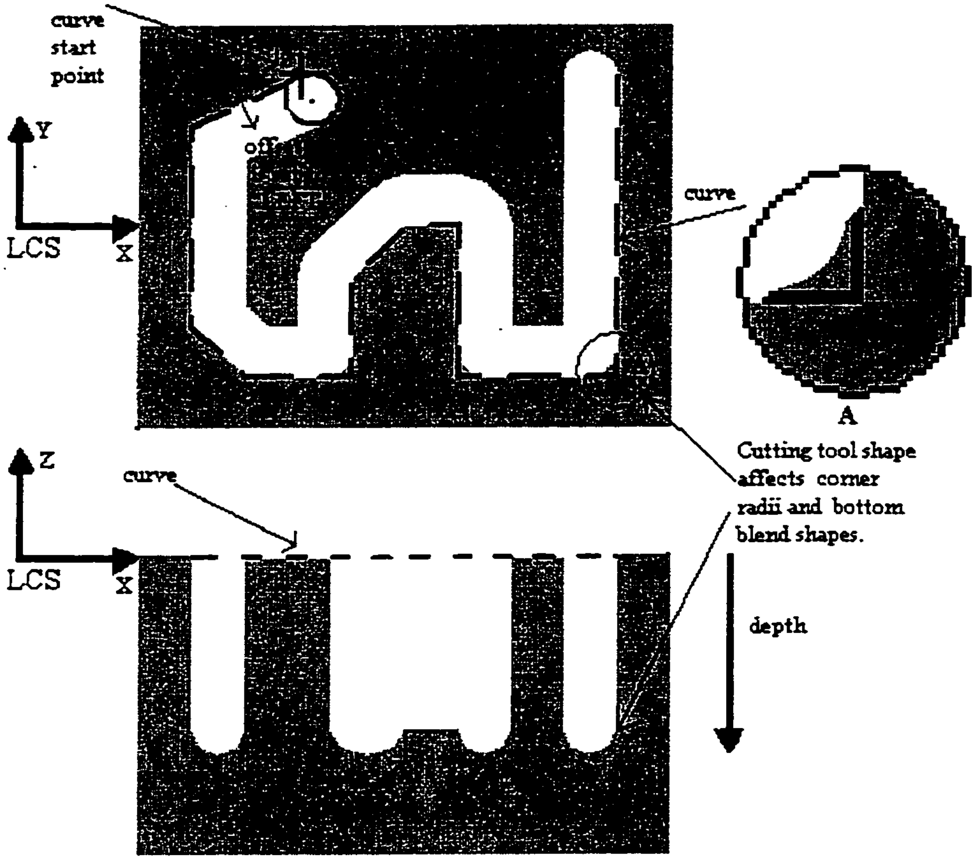
Element belonging to operation-hole	Description
Point	Hole location
Depth	Cutting depth. Positive real number

Attribute of operation-hole	Type & description
Sequence-number	Text, identification of this operation that employs process-planning information.
Tolerance-set-id	Text, reference identification for the applied tolerance-set element in tolerance-lib element

Note: The attributes above are common for all operation elements, so these will be skipped in the other features.

Figure 4.5 Operation-hole element

- Contour Feature (*operation-contour*) - Figure 4.6.



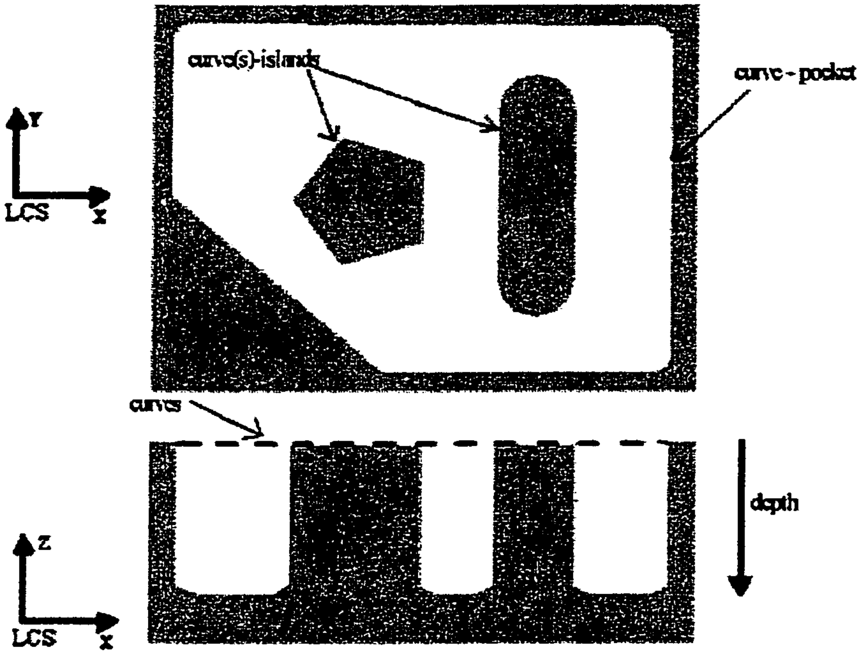
Elements	Description
Curve	A closed or non-closed curve for a cutting tool trajectory
Depth	Cutting depth. Positive real number
Offset	The value has one of "NONE","LEFT","RIGHT". This element determines the horizontal tool location according to the curve direction.

Figure 4.6 Operation-contour element

The operation-contour can create a shape resulting from sweeping a cross section (which corresponds to a cutting tool section chosen for this operation) along a curve. Machining operations commonly called slot, groove, edge, profile or channel may belong to the *operation-contour* category. Depth and offset direction are used to locate the tool cross section in each of the vertical and horizontal directions. In case the offset is set to "right" (or "left"), the tool cross section follows the contour curve with its left (right) side contacting the curve geometry if seen in the top orthographic view. The offset value

"None" means that the tool section creates a sweeping volume by following its center along the curve geometry. The actual cutting tools used to create these features include flat end mills, tapered end mills, ball end mills, side mills, round-end mills, along with an almost unlimited variety of specialty shapes like t-slot, chamfer and dovetail cutters. The combination of using simple geometry to define the contour, along with the virtual tool definition provides both power and versatility with a system that is conceptually simple.

- Pocket (operation-pocket)-Figure 4.7



Elements	Description
Curve(s)	One or more curve. The first curve is used for pocket boundary and the others for islands
Depth	Cutting depth. Positive real number

Figure 4.7 Operation-pocket element

The pocket geometry is composed of one or more closed curves and a pocket depth. In the *operation-pocket* element, the curves are ordered to indicate which curve is the pocket boundary and which is the island geometry. The first curve defined in the *operation-pocket* is assumed to be the external pocket boundary and the others as island geometry. Curve boundaries may not intersect, and island boundaries may not lie outside the pocket boundary. The z-level of all curves is currently assumed to be the same, but a future enhancement could allow different z-levels for the islands. A closed volume is created by sweeping closed curves in the z direction by the value of the depth attribute. The virtual tool shape can be used to define non-vertical walls, i.e. a tapered end mill would create a wall with a draft angle. A through pocket is defined by specifying a depth that goes all the way through the workpiece.

There are two more volume removal operation elements in NCML, which are derivatives of the pocket operation. These are *operation-boss* and *operation-facing*. The attributes of operation-boss and operation facing are exactly the same as those in the operation-pocket element.

- Boss Feature (operation-boss) –see Figure 4.8

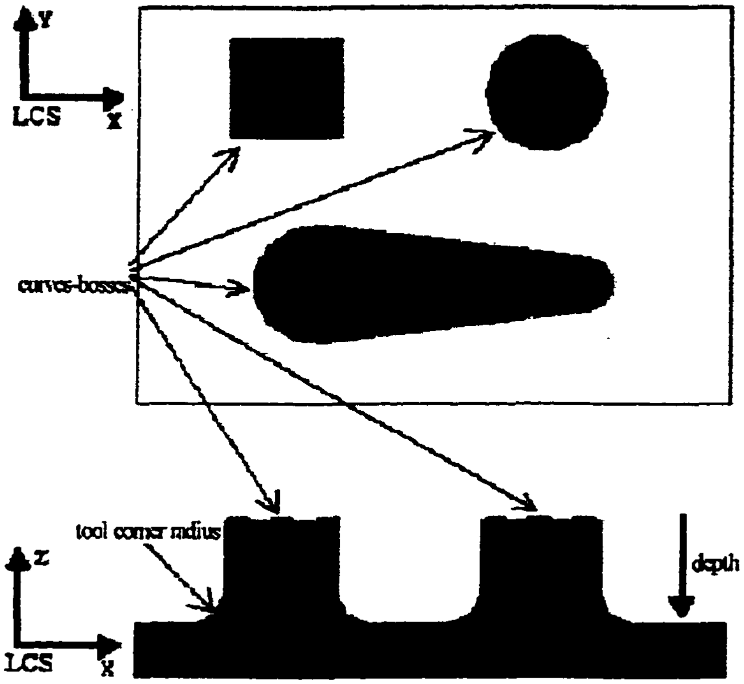
In NCML, the boss operation is assumed as a pocket with one or more islands and an exterior boundary defined by the work piece limits. As seen in Figure 4.3, all material around the boss (islands) curves is removed down to the level defined by the depth attribute.

- Facing Feature (operation-facing) – See Figure 4.9

The *operation-facing* is also a derivative of the operation-pocket. An *operation-facing* is a pocket with its external boundary also defined by the work piece limits, but with no island geometry.

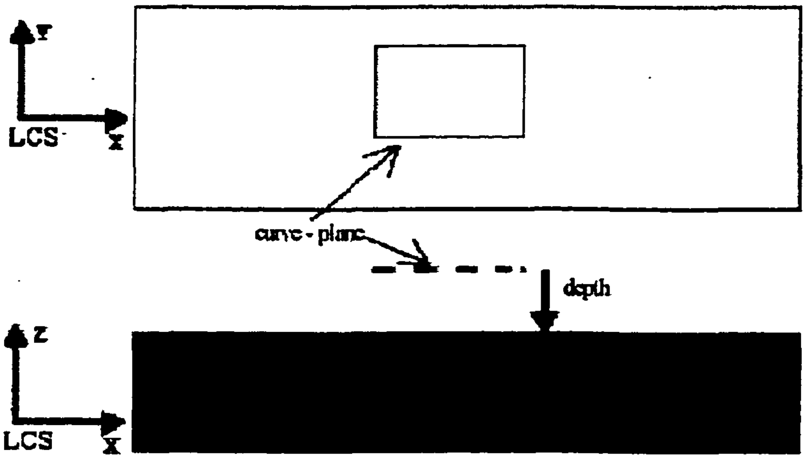
Some examples which illustrate the utility of the virtual tool concept are now presented along with their associated NCML codes. In Figure 4.10, one can see the feature definition in NCML and graphical examples of the features.

Figure 4.10-(a) shows an operation-hole element. The hole on the left is a simple hole, however, the second one is a hole with a countersink. Figure 4.10-(b) shows two slot cuts, which are identical except for the tool definition. Notice the difference of corner radii in the simple pocket operation in Figure 4.10-(c).



Elements	Description
Curve(s)	One or more curves. All of these curves are used to define boss geometry.
Depth	Cutting depth. Positive real number

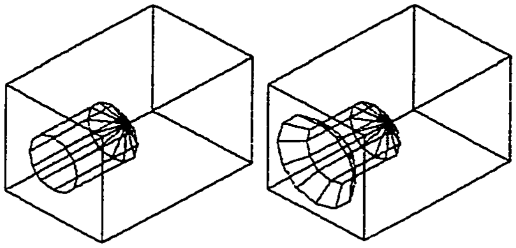
Figure 4.8 Operation-boss element



Elements	Description
Curve(s)	A closed curve is used to define a plane to set the height of cutting plane along z-axis in LCS.
Depth	Cutting depth. Positive real number

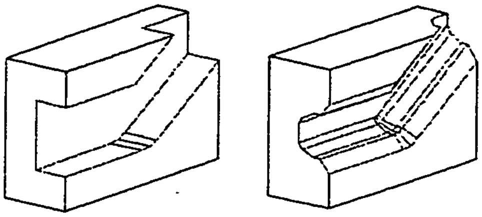
Figure 4.9 Operation-facing element

```
<operation-hole seq-no="S:1-PH:1-OP:1">
        <depth value = "1.50"/>
        <point x="1.50" y="1.00"/>
</operation-hole>
```



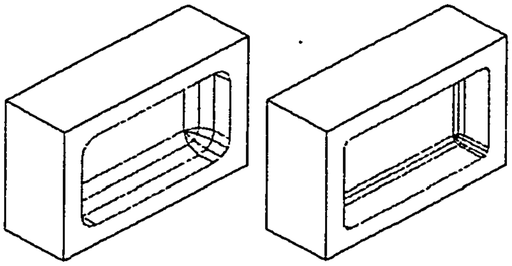
(a) operation-hole codes with two different tools

```
<operation-contour seq-no="S:1-PC:1-OP:1">
 <depth value="0.50"/>
  <offset direction="NONE"/>
  <curve>
   <line>
    <point y="1.00"/><point x="1.50" y="1.00"/>
   </line>
   <line>
    <point x="3.00" y="2.00"/>
   </line>
  </curve>
</operation-contour>
```



(b) operation-contour codes

```
<operation-pocket seq-no="S:1-PP:1-OP:1">
 <depth value = "0.50"/>
 <curve>
  <line>
   <point x="0.30" y="1.70"/><point x="0.30" y="0.30"/>
  </line>
  <line>
   <point x="2.70" y="0.30"/>
  </line>
  <line>
   <point x="2.70" y="1.70"/>
  </line>
  <line>
   <point x="0.30" y="1.70"/>
  </line>
 </curve>
</operation-pocket>
```



(c) operation-pocket codes

Figure 4.10 Pictorial examples of the virtual tool usage for NCML operation elements

4.6 Tool Characterization & Modeling in NCML

The cutting tool geometry determines the actual machining volumes created by the operation elements in NCML. Cutting tool geometry is defined in the tool element under the tool-set element.

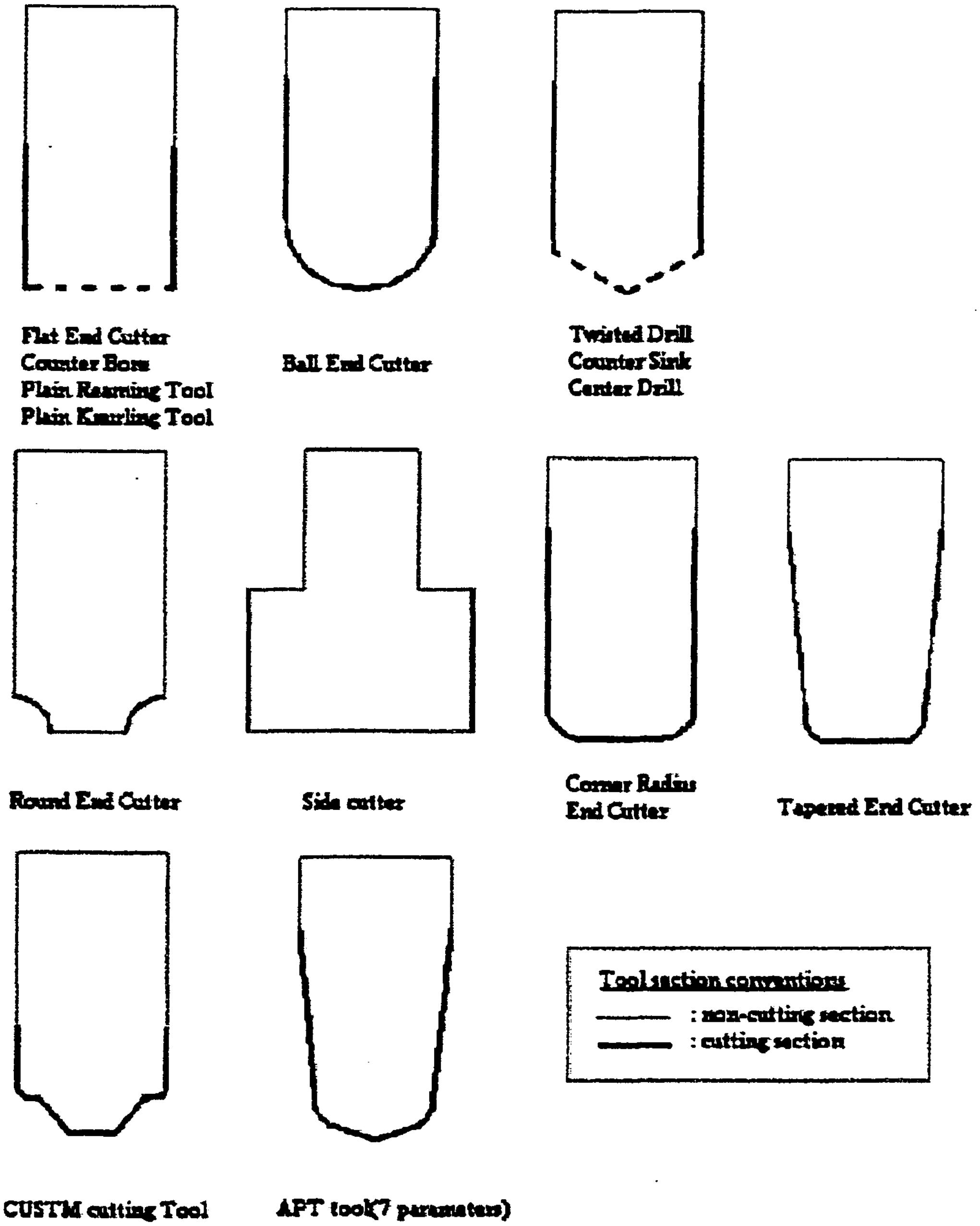


Figure 4.11 NCML cutting tool shapes

The tool element can express a variety of cutting tool shapes; basically any cross section that can be expressed as a curve consisting of line and arc segments. Future enhancements could expand the available geometric primitives beyond lines and arcs. Tool shapes are generalized and categorized into seventeen generic tool types[6]. Figure 4.11 shows some typical types of tool shapes that can be expressed in the *tool* element. Refer to Appendix B for details.

Machine tools are limited to 3-axis CNC milling machines here, since currently only manufacturing features that can be achieved by 3-axis CNC milling machine are included in NCML. However, machine tool information is unnecessary in NCML, because the choice of the machine tool is solely up to manufacturing. Of course, design specifications may limit the feasible machine tools because of part size limitation or tolerance requirements.

4.7 NCML and STEP-NC

STEP-NC[7] has been developed as a high level language, which facilitates operating NC machines. It is currently a draft standard being developed by Technical Committee 184, Subcommittee 4 (TC184/SC4). Since STEP-NC seems to share many of the same goals as NCML and uses approaches analogous to NCML, this section attempts to compare the two approaches and notes the differences. However, since STEP_NC is a fairly recent development, only a small amount of information is available and there are no commercial implementations yet, it is quite difficult to make a detailed comparison between the two approaches. The STEP-NC information in this section can be found from websites (www.steptools.com, www.step-nc.com and www.step-nc.org/) and the following references [SNC_P01, SNC_P10, SNC_P11 and SNC_P111].

4.7.1 STEP-NC: Description and Benefits

STEP-NC (ISO 14649) is a new model for data transfer between CAD/CAM systems and CNC

---

[6] See Appendix B - to see each type in detail.

[7] The International Standard "Data model for computerized numerical controllers"(ISO 14649).

machines. It is intended to replace ISO 6938 (commonly known as G-codes) because of the portability limitations[8] of ISO 6938. By employing the concept of Workingsteps, STEP-NC specifies machining processes rather than machine tool motion. Workingsteps correspond to high-level machining features and associated process parameters. The CNC machine controller is responsible for translating workingsteps into axis motion and tool operations that are conventionally described with ISO 6938 [SNC_P01].

STEP-NC is composed of three different but mutually related components. These components are geometry description, technology description, and sequenced task description. Workingsteps, that are components of a sequenced task, are executed as machining operations in linear order. However, the descriptive level of tasks in a workingstep is quite declarative, not detailed. An example of a workingstep would be the roughing of a pocket or the finishing operation of a freeform surface region. The detailed information such as tool data, machining functions, machining strategies and other process data of the workingstep is referenced from the technology description and geometric definition. For example, machining feature geometry and free-formed surfaces for each workingstep are referenced from the geometric definition.

STEP-NC uses International Standards generally called "STEP"(ISO 10303) for describing its information. All geometry data in geometric description such as workpieces, setups, and manufacturing features are described using the ISO 10303 data format. The following sentences are quoted form [SNC_P01-p4] to explain how STEP-NC is related to STEP.

> "The design phase results in CAD data (Geometry ISO 10303 AP203) and includes definition of all the part features in ISO 10303 AP224. The process planning phase generates the resource requirements for part fabrication, using ISO 10303 AP213, and other results suitable for use in a Manufacturing Execution systems (MES). Process planning also splits the AP224 manufacturing features into sets suitable for various processes, e.g., milling turning, EDM and inspection (which also uses AP210). The AP224 feature sets are used during the computer-aided manufacturing (CAM) phase, which generates ISO 14649 files that are executed by the controllers."

STEP-NC files are represented according to ISO 10303 Part 21(Implementation Method: Clear text encoding of exchange structure). STEP-NC has the potential to dramatically change the process of

---

[8] There are three major reasons. First, ISO 6983 focuses on programming the tool center path with respect to machine axes, rather than the machining process with respect to the part. Second, the standard defines the syntax of program statements, but in most cases leaves the semantics ambiguous. Third, venders usually

fabricating machined parts. First, the tool path generating function, which is typically done by CAM systems, will be assumed by the CNC. This requires that CNCs should control machine tools directly from 3D data in STEP-NC files. Second, the data flow from design to fabrication will be streamlined by adopting the International Standards based on STEP. This will eliminate undesirable data exchanges between CAD systems and CAM systems, which are a frequent source of problems in the conventional method. Third, STEP-NC will make it possible to store an integrated model that includes both fabrication and geometric design information. This integrated model can be regarded as a universal reference model not only for design but also for machining. Local geometric modifications in design data (e.g. changing the diameter of a drilled hole) can be accomplished without changing the rest of the STEP-NC file. Therefore, the fabricator may use the modified STEP-NC model with a minimum of effort. This will really improve the reusability of CAD/CAM models.

4.7.2 Comparison of NCML with STEP-NC

In common with NCML, STEP-NC is designed to provide higher quality information to the fabricator and to facilitate bi-directional communications between the design and fabrication sides. However, the philosophies of STEP-NC and NCML are somewhat different. STEP-NC attempts to provide full descriptions of all machining activities such that CNCs can create tool paths and can execute machining operations from STEP-NC information, while NCML leaves most of the details of the machining activities to the fabricator. This most significant difference is caused by the different purpose of each data format. NCML is devised to achieve the clean interface between the design and the fabrication and is not a fabrication side model intended to facilitate machining operations executed by CNCs. Although NCML is conveying design information in terms of machining features, NCML is intended to deliver accurate design intentions and to accelerate the machined part fabrication process.

This basic difference in purpose results in the following differences in implementation.

- NCML is concise. The most significant advantage of NCML over STEP-NC is the relative simplicity of NCML. Because of the different goals of NCML and STEP-NC, an NCML file would include less

---

supplement the language with extensions that are not covered in the limited scope of ISO 6983[SNC_P01].

fabrication information than STEP-NC.

- The "virtual tool" concept employed by NCML, which is not included in STEP-NC, also leads to a more concise definition of machining features while also achieving greater versatility. NCML can represent all of the machining features included in STEP-NC despite the fact that it has far fewer types of feature types (five vs. more than twenty types and sub-types). For example, STEP-NC has five sub-types of swept profiles: square-u, round-u, tee, vee and general. NCML would represent all of these as simply a "contour" type feature and use the "virtual tool" concept to represent any of these shapes, as well as an unlimited number of other shapes that could not be represented in STEP-NC. Fillets and chamfers, which exist as separate features in STEP-NC are represented in NCML as contours with an associated tool of the correct shape.
- NCML is suitable for the Internet based commerce and applications. First, NCML relies exclusively on XML, the data standard gaining acceptance on the web, while STEP relies on Express. Since STEP originated long before the emergence of the web, it is understandable that it would not be based on XML. Second, NCML can take advantage of essentially free, but useful utilities available on the Internet. Using NCML, an Internet compatible graphical model in VRML can be easily obtained. This graphical model gives a better understanding of the design intent in NCML.
- The relative simplicity of NCML makes it possible to develop additional utilities that have the potential to enhance Internet based commerce. Quotation Helper, an example of such a utility, is described in Chapter 5 of this dissertation. This utility makes it possible for fabricators to prepare bids for RFQs that are consistent and easily generated directly from the NCML. STEP-NC does not currently have any similar capability.
- NCML is appropriate for low technology environments. Although STEP AP224 (one of the primary protocols of STEP-NC) has been around as a standard for a number of years it has yet to be adopted by industry. Moreover, since most job shops rely on paper drawings, it seems doubtful that STEP-NC can be a practical solution in such a low technology environment. NCML is easily created from paper drawings and can be used as a fabrication start medium.
- The NCML data structure could be read by conventional CAM packages (e.g. MasterCam,

FeatureCam, etc.) while STEP-NC places most of this functionality into the CNC controller. This puts a constraint on machine tool builders that may be difficult while also imposing a radical change in the modus operandi of users.

As a conclusion, summarized comparisons between STEP-NC and NCML are given in Table 4.1.

Table 4.1 Comparisons of NCML with STEP-NC

Criteria	NCML	STEP-NC
Goal	Clean design data interface between the design and the fabrication of custom machined parts	Replace G-Codes with a higher level language and integrate fabrication data with design data
Data rep. language	XML	EXPRESS
Design contents	Machining features, virtual tools	Machining features (STEP AP224)
Editable by hand	YES	GUI software level
Internet compatibility	YES	Possible. Transferring the EXPRESS format to the XML format has been done
Task organization in the structure	Grouping and linear ordering are used in its hierarchical structure	Linear order
Task contents	High level information based on the machining features	Detailed lower level machining steps which seems to be analogous to the UMO concept used in this research (see Section 5.3.3)
Tolerance	Simplified tolerance scheme currently	STEP can handle tolerance information.
Fixture information	Setup. Fixture may be implemented soon.	Setup and Fixture
Detail machining Strategy	Not included	Included
Implementation	Relatively easy	Complicated. Should consider all relevant STEP standards and clear all conflicts among currently developing standards
NC Machining	Simplified Process planning and post-processing is needed to turn to machining commands	Intend to use NC machining directly. In terms of the higher-level language replaces G-codes, STEP-NC contains more information.
Design data integration	Needs design data interchange	Integrated with STEP design data
Finished part geometry	Implicit	Referred to STEP design data
International Standard	No	Evolving standard
Appropriate business type	Job Shops	Large manufacturers
E-Commerce Possibility	YES	Possible.
Intelligent CNC needed to be executed	Preferable. Mainly relies on its programmable capability	Essential. To interpret STEP-NC data to machine axis movements intelligently.

CHAPTER 5

SOFTWARE SYSTEM IMPLEMENTATION

This chapter describes the functional aspects of the software systems developed in this research: FACILE/Design, FACILE/Fabricate and E-Mill. Each function was developed with the intention of being a part of the NCML based fabrication process. This chapter shows how each function is implemented and what users can do with the functions.

The explanations of the functions are sequenced according to the working orders of the software systems and the NCML work procedure. Readers can refer to Figure 3.2, 3.3 and Table 3.1 to see the role of each function in the process and how each one is related to others in the work process.

5.1 FACILE/Design

FACILE/Design is the design side software system in the proposed NCML based part fabrication process. This is a CAD software system that can be used to create NCML files. While NCML generation is the essential function, other supportive functions are necessary for embodying machined part information smoothly in FACILE/Design.

The following functions in FACILE/Design will be discussed in this section:

- Modeling a part, and editing information
- Macro Simulation
- NCML and VRML file generation

5.1.1 Machined Part Modeling

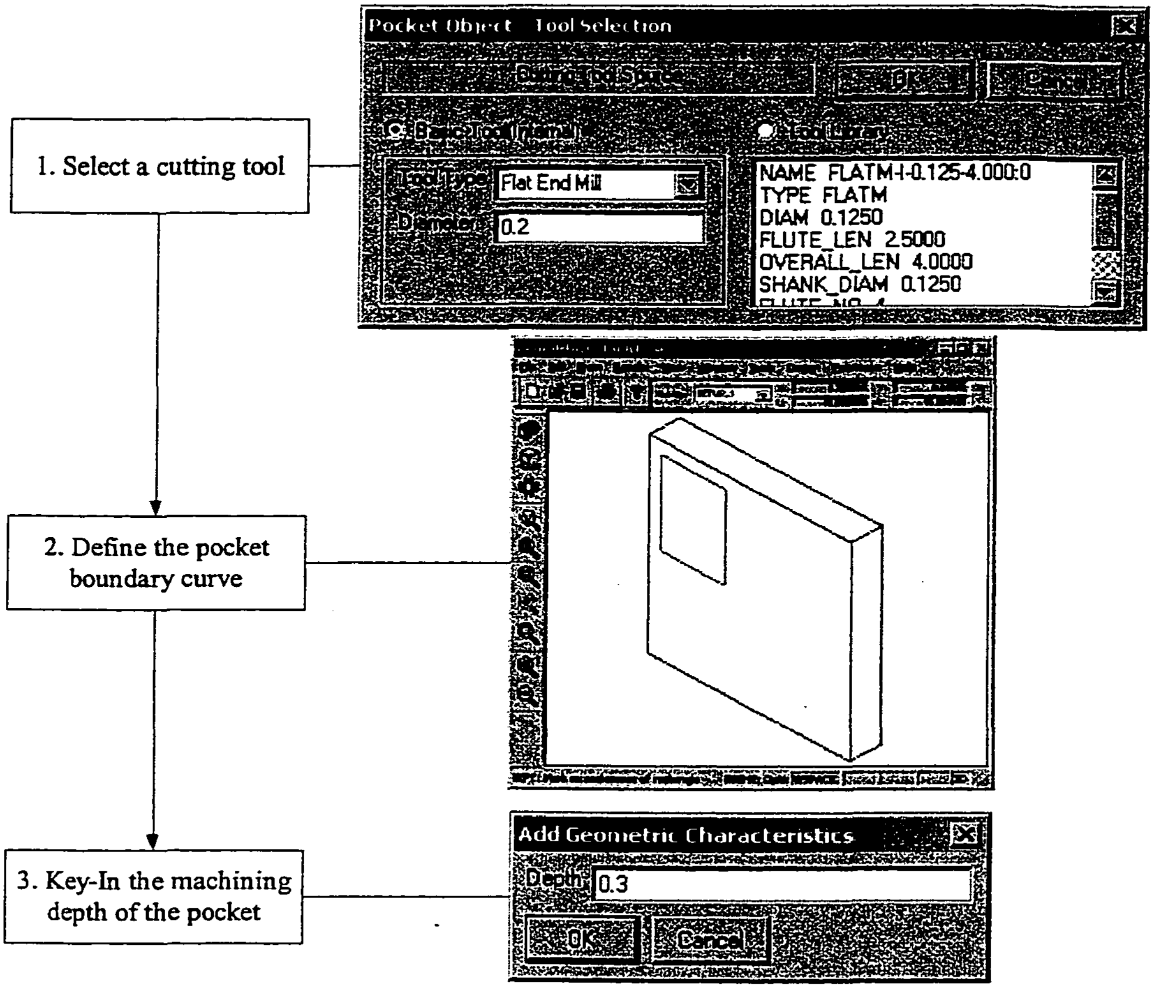
The functional goal of FACILE/Design is to generate NCML documents which describe machined parts, and which are syntactically consistent with the XML standard.

NCML can be edited and created with any text editor. However, editing geometrical information with text editors is not only extremely tedious, but is also error-prone. The modeling functions in FACILE/Design can be characterized as follows:

- Design by Feature. FACILE/Design uses the "Design by Feature" approach in its part modeling. Users can create a machined part model by defining manufacturing features (machining operations) from the standpoint of the designer. The user inputs the machining feature definitions into NCML. For example, a hole feature can be defined with hole location, depth and shape of the cutting tool used. The features in FACILE/Design are organized into *patterns* and *setups* in the hierarchical part structure of NCML.
- Graphical User Interface. Like other current CAD systems, FACILE/Design uses a GUI (Graphical User Interface). Basic functions like zooming, rotating, and scaling are utilized. Geometrical elements, like points, lines, curves, etc., can be defined in 3D space by selecting existing geometry or keying in coordinates.
- Geometry editing capability. Basic modeling functions in FACILE/Design include geometry editing functions (copy/move the feature geometry, change the location), and feature attributes editing capabilities (change the depth or offset).

Figure 5.1 illustrates the procedure for defining a pocket operation in FACILE/Design and Figure 5.2-(a) shows how geometry-editing functions lead to part shape changes.

The order of feature creation results in a simple process plan as illustrated in Figure 5.2-(b). FACILE/Design users can edit the machining sequence by changing the numbered part of each operation name, which is numbered according to the creation order. This leads to changes in the process plan delivered to the fabricator.



1. Select a cutting tool

Tool Type: Flat End Mill

Diameter: 0.2

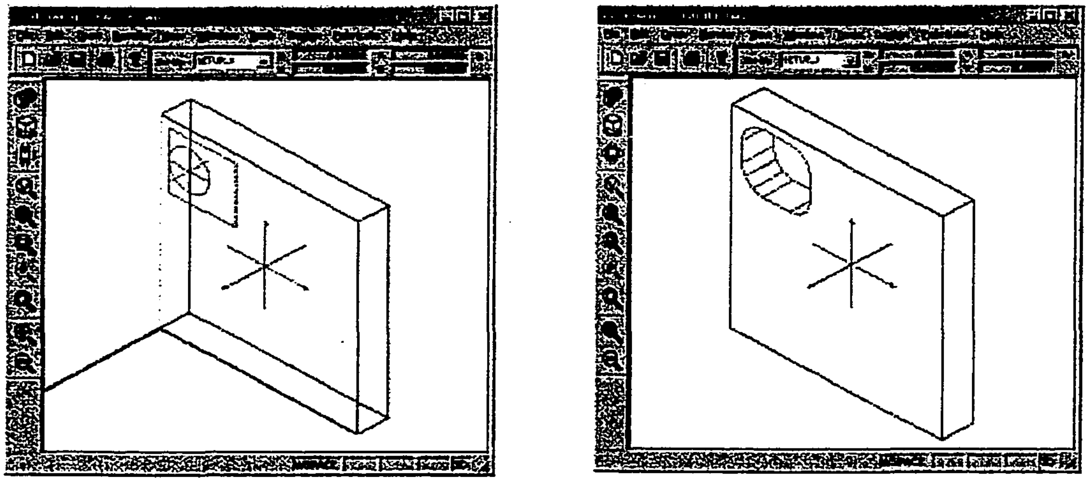
```
NAME  FLATM-1-0.125-4.000:0
TYPE  FLATM
DIAM  0.1250
FLUTE_LEN  2.5000
OVERALL_LEN  4.0000
SHANK_DIAM  0.1250
```

2. Define the pocket boundary curve

3. Key-In the machining depth of the pocket

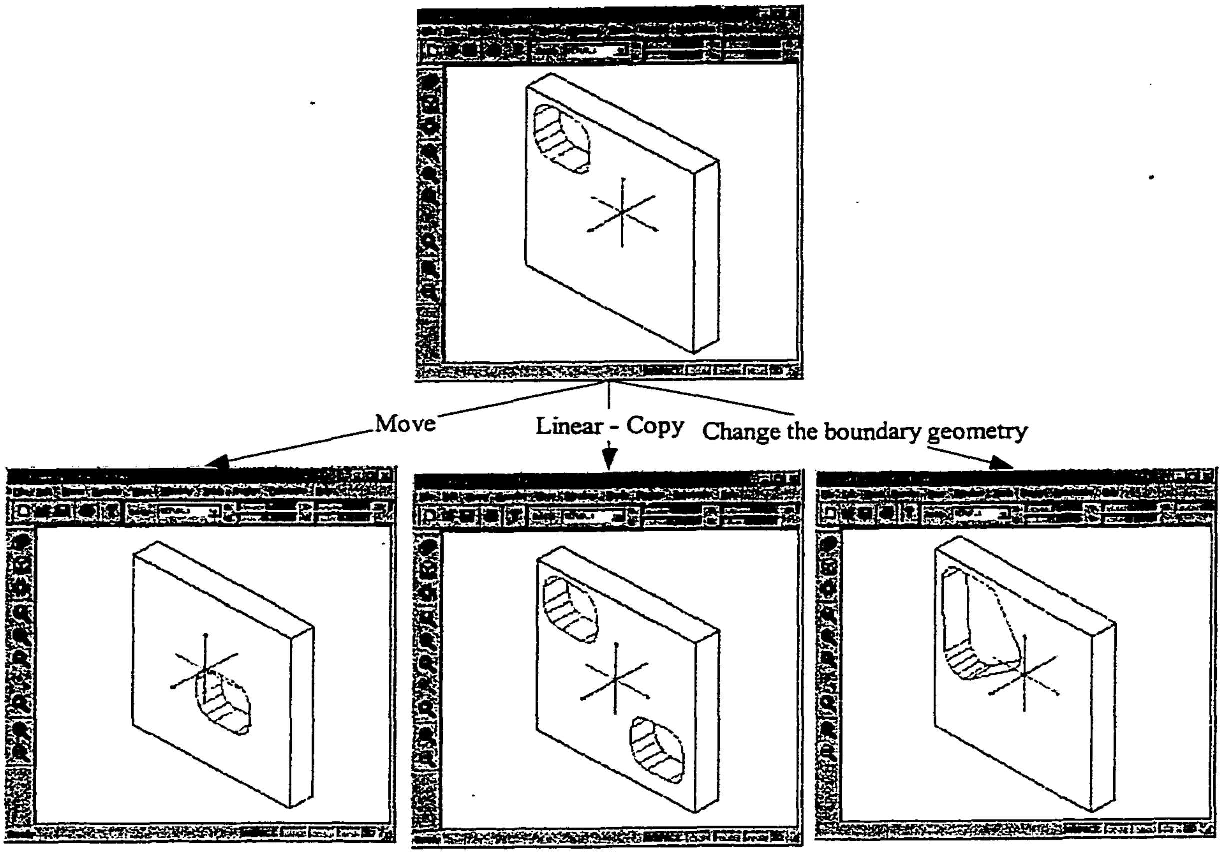
Depth: 0.3

(a) Define a rectangular pocket operation

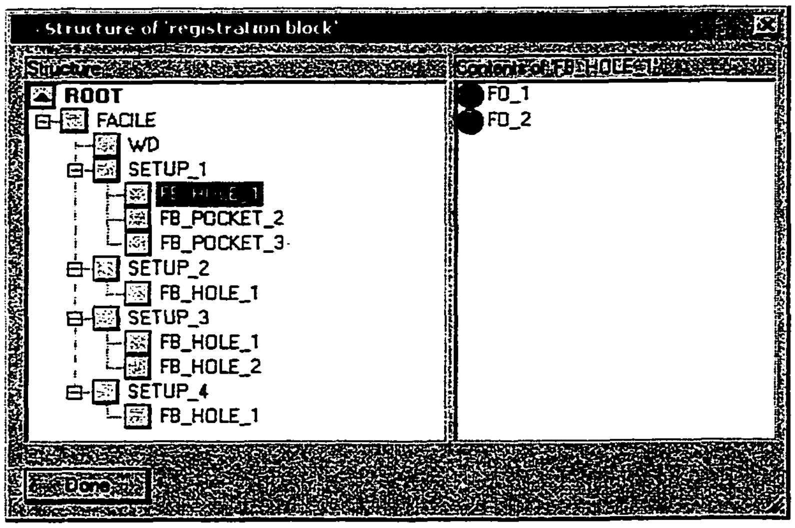


(b) A pocket shape in FACILE/Design (The left image is the pocket definition in a schematic view, the right one is the part shape after the pocket operation)

Figure 5.1 Pocket machining feature creation in FACILE/Design



(a) Feature editing functions
(Move, copy and geometric modification-move two corner points of the pocket boundary)



- ROOT
  - FACILE
    - WD
    - SETUP_1
      - FB_HOLE_1
      - FB_POCKET_2
      - FB_POCKET_3
    - SETUP_2
      - FB_HOLE_1
    - SETUP_3
      - FB_HOLE_1
      - FB_HOLE_2
    - SETUP_4
      - FB_HOLE_1

● FD_1
● FD_2

(b) Process editor - users can see the hierarchical structure of the designed part and change the process order

Figure 5.2 Editing functions in FACILE/Design

5.1.2 Macro Simulation

The swept volumes for the machining features are subtracted from the object workpiece in a step that is called *Macro Simulation*. This shows the user what the final part will look like. The same capability can be used by the Job Shop to easily visualize the sequence of operations and the resulting shape of the part before generating any tool paths.

Macro Simulation in FACILE/Design means that a function simulating machining operations is simple enough to provide meaningful information to the part designer without having to simulate individual tool paths (Micro-level Simulation). With very little effort, users can see how a part in NCML would be machined and how it would appear after each operation.

Since the NCML definition contains geometric information about the manufacturing features and cutting tool information, it is possible to create the volume of each operation. A boolean subtraction of these solid volumes from the workpiece results in the final solid intended in NCML.

Macro Simulation is a part of the FACILE/Design system and provides the following outputs:

- Simplified cutting simulation based on solid subtracting operations. Users can watch how a NCML part would be machined according to the operation order stored in the NCML data structure. With a little knowledge of machining, users may detect erratic cutting sequences or infeasible cutting operations and then suggest better ways to machine the part.
- Macro Simulation turns the original workpiece into a final machined shape. Investigating part geometry with zooming, rotating or measuring functions in FACILE makes checking feasibility easier.

Implementation

The flowchart for Macro Simulation is given in Figure 5.3. These procedures traverse every *operation* element in NCML according to the machining orders embedded in the hierarchical structure and calculate each operation solid volume. This is done with consideration for the type and geometry of the operation, the cutting tool information, and the setup orientation that are obtained from the structural element in the NCML file that contains the *operation* elements.

The methods of constructing machining volumes of NCML operations vary with the type of operation (hole, contour, or pocket). The hole machining volume is the simplest to obtain. (See Figure 5.4)

For the contour operation, the amount and direction of offset from the contour is needed. Based on the offset, create the half tool shape at the contour starting point (Figure 5.5- (b)), sweeping the tool section along the contour geometry and adding another half tool shape at the contour end point (Figure 5.5-(c)) provides the exact tool trajectory of the cutting volume. Merging intermediate solids and subtracting the merged solid from the workpiece give the resultant solid geometry for the contour operation (Figure 5.5-(d)).

The pocket operation can be achieved by combining solid operations used in both the hole and the contour operation cases. First, a lumped volume that is not affected by the different tool shapes can be generated by offsetting the pocket definition in the inward direction by as much as the tool radius, and linearly sweeping this profile in the negative z direction by cutting depth (Figure 5.6- (b)). Secondly, as done in the Contour feature, sweeping the tool section along the bottom edges of the lumped solid volume gives the geometry of pocket walls (Figure 5.6- (c)). Merging the intermediate solids constructs a solid being machined by the Pocket operation (Figure 5.6- (d)). Figure 5.6- (e) shows the final shape of Pocket in Macro Simulation.

Example

Figure 5.7 shows the screen shots of macro simulation applied to an NCML model that contains a single setup and a variety of operations with different shapes. In Figure 5.7 (a) the schematic drawing in FACILE shows what the initial model looked like, the simulation process of the machining volumes being removed, and the cutting tool movements (Figure 5.7 (b)). Finally the result of Macro Simulation is shown in Figure 5.7 (c)

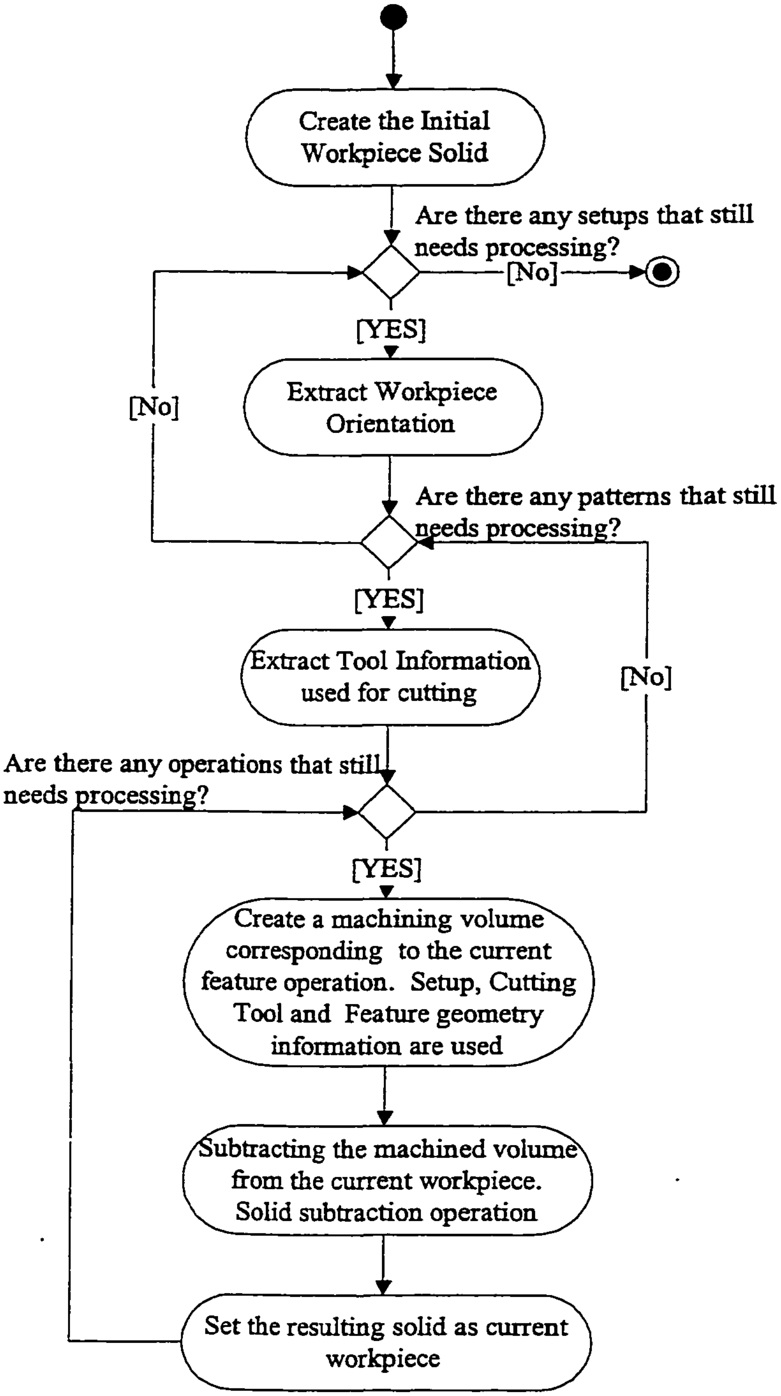
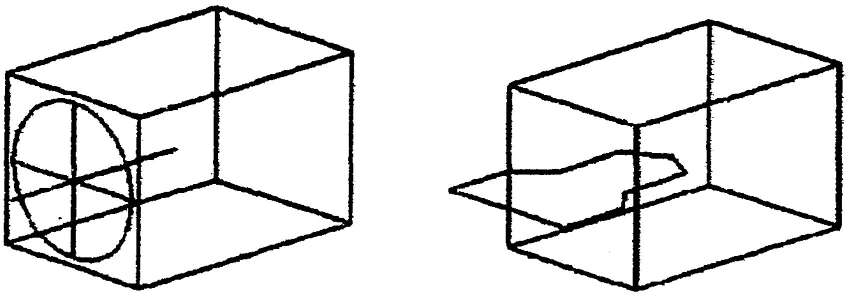
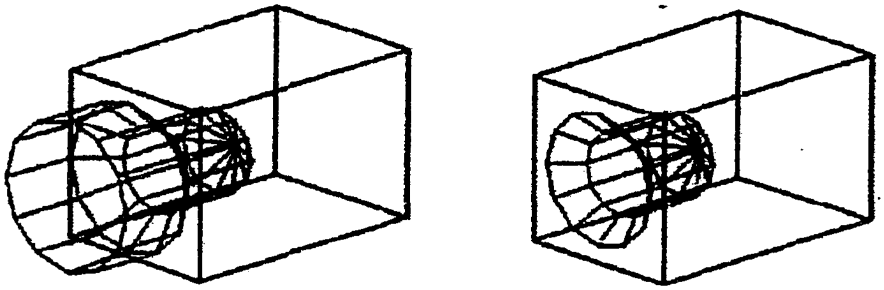


Figure 5.3 Flowchart of Macro Simulation



(a) Hole feature symbolic representation in FACILE

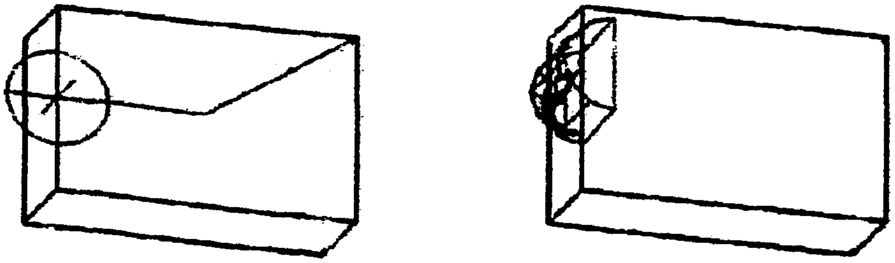
(b) Place the cutter section at the hole position



(c) Rotate the section around the cutter center by 180 degrees to get a tool shape solid

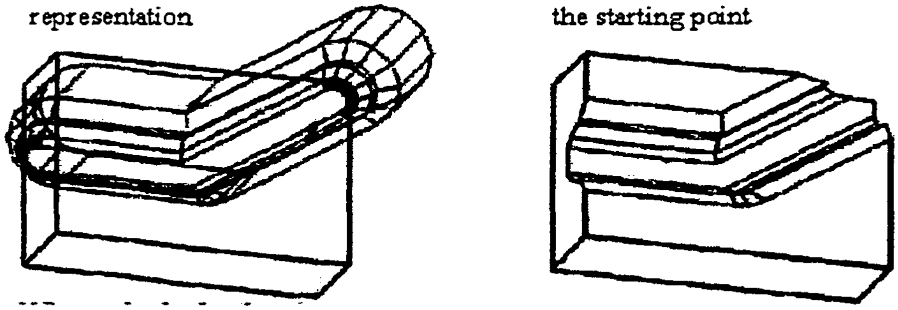
(d) Final shape - subtracting the tool solid from the workpiece

Figure 5.4 Macro Simulation- hole operation



(a) Contour feature symbolic representation

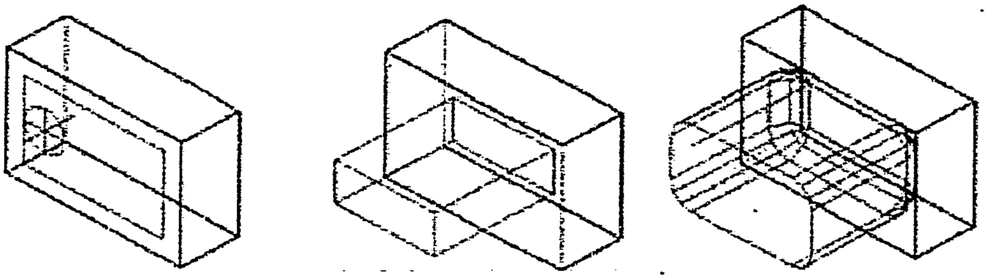
(b) Place the half tool solid at the starting point



(c) Sweeping the tool section along the curve leads to a tool trajectory solid

(d) The final shape of contour macro simulation

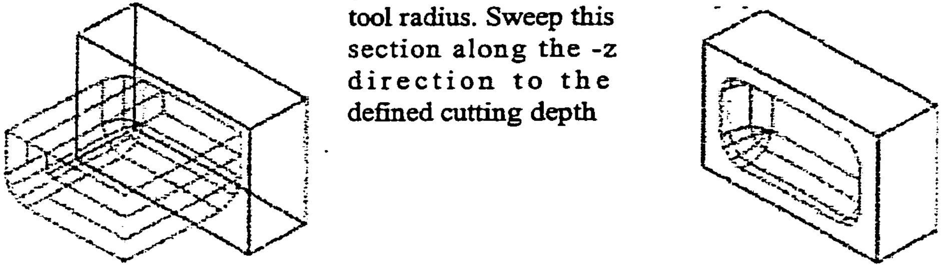
Figure 5.5 Macro Simulation- contour operation



(a) Symbolic pocket definition

(b) Offset pocket geometry inward by tool radius. Sweep this section along the -z direction to the defined cutting depth

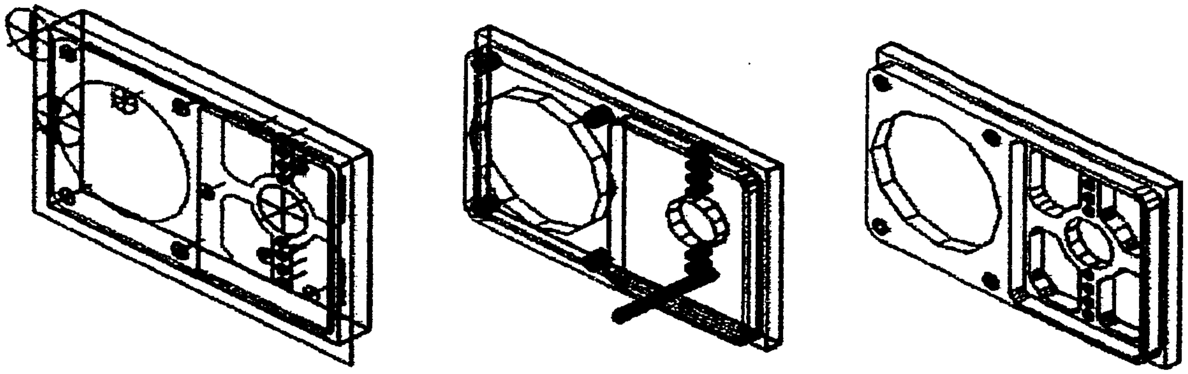
(d) Machining volume



(c) Sweep the tool section along the bottom edges

(e) Final shape of the pocket operation

Figure 5.6 Macro Simulation- pocket operation



(a) Symbolic feature representations before Macro Simulation in FACILE

(b) Screenshot in Macro Simulation

(c) Final part shape created from Macro Simulation

Figure 5.7 Macro Simulation illustrations

5.1.3 File Generation

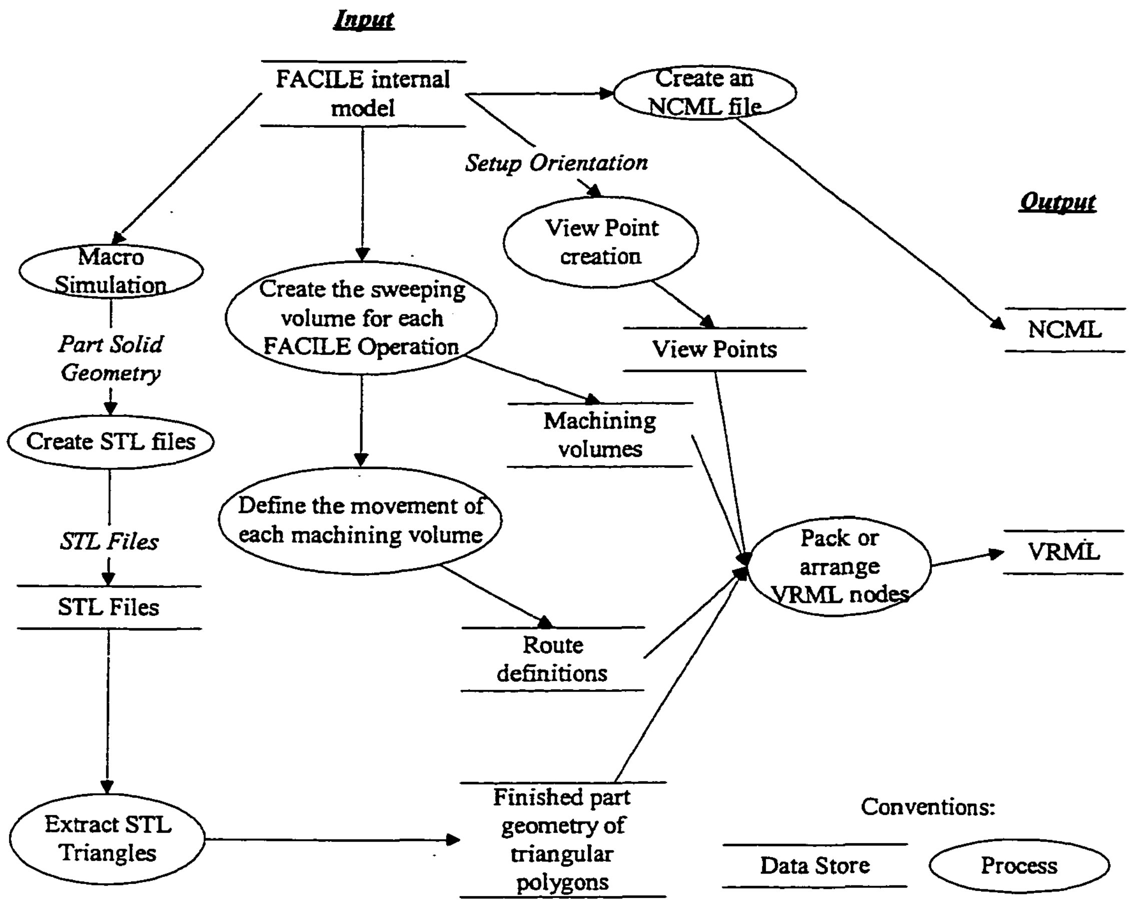


Figure 5.8 Internal procedure for file creation in FACILE/Design

The results of FACILE/Design are two document output files. One is the NCML file and the other is the VRML file for graphical information display on the E-Mill web site. These files will be registered as information related to the posted part RFQ on the E-Mill e-commerce web site.

Generating an NCML file from the FACILE/Design model is simple enough, since the internal data structure of the FACILE/Design model is identical to the NCML hierarchical structure. Therefore, the NCML generator creates a NCML document by referring to the NCML DTD and extracting data from the FACILE/Design model. The NCML correctness is checked on the DTD and any grammatical errors are avoided by adopting a validating XML parser. The Xerces C++ XML parser of the Apache XML project (http://xml.apache.org/) is used to create NCML files in FACILE/Design.

Showing a graphical representation with the NCML document helps users to visualize the part.

For the purpose of showing the 3D geometric model on the E-Mill web site, FACILE/Design can create a VRML model from the NCML document and from the boundary representations of the solid geometry that are created as a result of Macro simulation. With the generated VRML model for the machined part, users can display the graphical model on a web page and visualize each machining operation by clicking on the machining volumes corresponding to each machining operation. This active characteristic is attained by using VRML route and anchor nodes.

The detailed process of creating output files from the FACILE/Design system is explained with the process diagram in Figure 5.8. Four basic elements are incorporated into the VRML file. The finished part geometry can be achieved in polygonal representations after Macro Simulation. Polygons can be easily changed into a VRML "IndexedFaceSet" node. Each machining operation volume can be represented as a VRML shape node, like "extrusion". When users click this shape with the mouse in the VRML viewer, the "Anchor" node is attached to move the machining operation backward and forward. In the route definition, the trajectory of the moving solid is defined in the last component, pre-defined view which is identical to the setup orientation in NCML, is attached as "viewpoint" nodes in VRML. This will help the user with returning to the default view anytime. An example of using the NCML and VRML output files is illustrated in the next section which describes the E-Mill application.

5.2 E-Mill

E-Mill is a vertical (focused on a specific kind of business or organization) B2B e-commerce site where buyers and sellers (job shops) of machined parts conduct e-commerce. By playing a middleman role in the matchmaking procedure, E-Mill also attempts to keep both buyers and sellers loyal to E-Mill by providing valuable services. The NCML format created in FACILE/Design is used as the standard part representation format in E-Mill.

5.2.1 Overall Site Organization

As seen in Figure 5.9, the E-Mill web site is composed of several web pages. It is also equipped with some auxiliary pages like Help and Download pages. Brief descriptions of the component web pages follow:

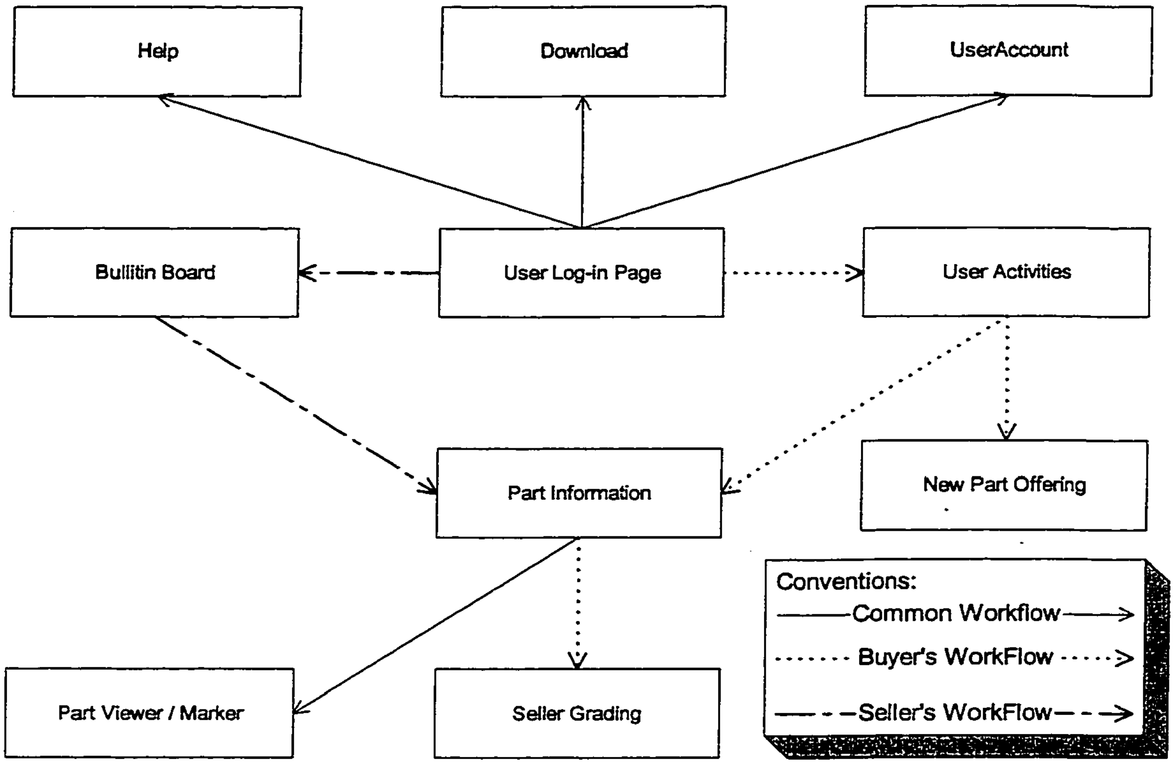


Figure 5.9 E-Mill web site organization diagram and workflow

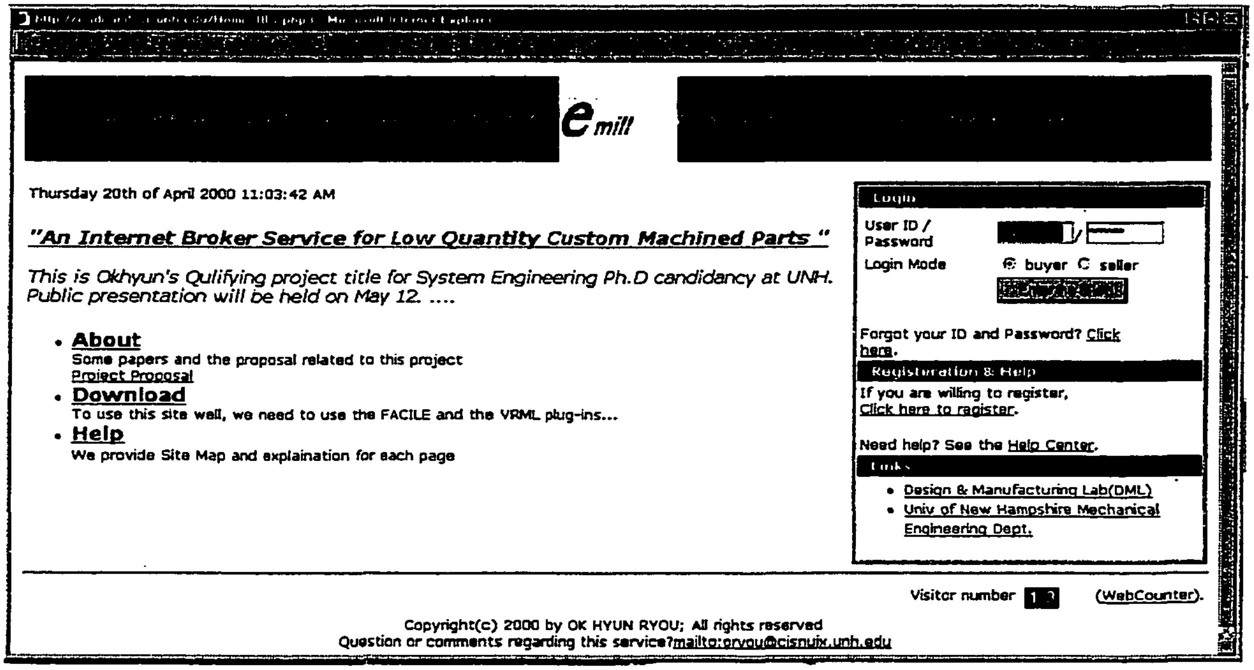
- **Home or User Login page**: The user can log in to the E-Mill system after user verification. A new user can register at this page. In addition, brief descriptions about E-Mill and some related URLs are provided.

- **Bulletin Board Page**: The list of parts offered by buyers can be brought up for sellers. A seller can sort the open parts for bidding in the order of part due date, part price estimated by the buyer, and buyer name. Clicking a part number leads the user to the "Part Information" page for the detailed investigation.

- **Part Information Page**: Users can examine all the information about a part offered by a buyer. Sellers can bid for the part at this page and the part buyer can compare bids on the part and can choose the bid winner.

- **User Activities page**: The contents of this page change according to user activities in E-Mill. The user can see the categorized part lists. For example, the part buyer can list parts according to part status (OPEN, CLOSE, FINISHED) and a seller also can list his own parts in the order of part status and bidding results.

- **Download page**: E-Mill customers can download the FACILE software and other required software (VRML plug-ins, graphic file viewers).
- **New Part page**: A buyer can post a new RFQ on this page. Considering the importance of this activity, the buyer should pass a user verification test when submitting a part order. A buyer can set part parameters and upload virtually any files related to the part RFQ.
- **Seller Grade page**: Buyers can see the other buyers' opinions about the specific seller. Moreover, a buyer can grade the seller using five distinct levels. Information about job shop certification may be found on this page.



Thursday 20th of April 2000 11:03:42 AM

"An Internet Broker Service for Low Quantity Custom Machined Parts "

This is Okhyun's Qulifying project title for System Engineering Ph.D candidancy at UNH. Public presentation will be held on May 12. ....

- **About**
  Some papers and the proposal related to this project
  Project Proposal
- **Download**
  To use this site well, we need to use the FACILE and the VRML plug-ins...
- **Help**
  We provide Site Map and explaination for each page

Login

User ID / Password

Login Mode ● buyer ○ seller

Forgot your ID and Password? Click here.

Registration & Help

If you are willing to register, Click here to register.

Need help? See the Help Center.

Links

- Design & Manufacturing Lab(DML)
- Univ of New Hampshire Mechanical Engineering Dept.

Visitor number 1 ? (WebCounter).

Copyright(c) 2000 by OK HYUN RYOU; All rights reserved
Question or comments regarding this service?mailto:oryou@cisunix.unh.edu

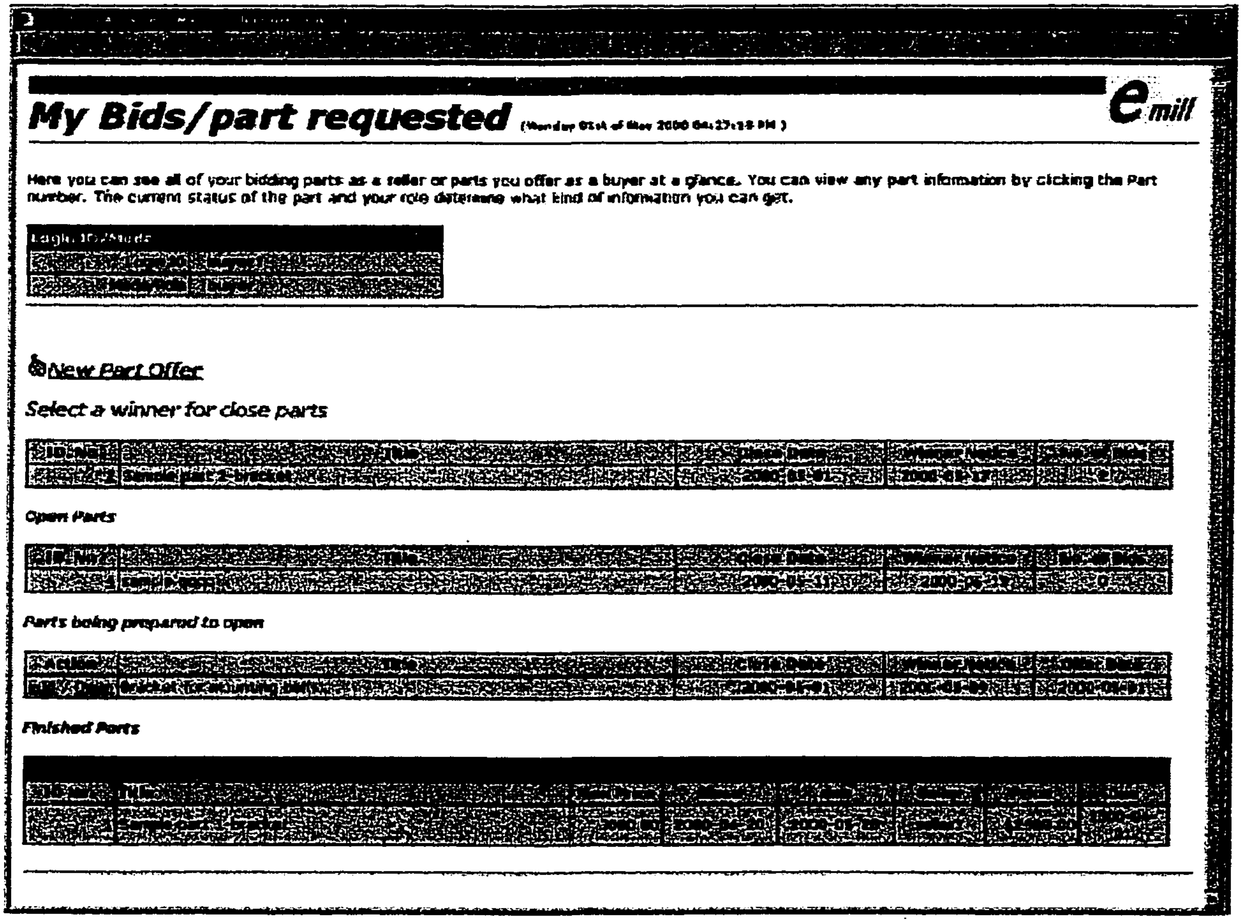
Figure 5.10 E-Mill home page snapshot

5.2.2 Home Page-Login Page

Figure 5.10 is the E-Mill starting page. From this page, the user can move to other functional pages. The main purpose of this page is user validation – only registered users can move to E-Mill functional pages.

5.2.3 User Activities Page

The web page in Figure 5.11 provides customized contents, which vary according to user activities. A buyer can see the list of parts sorted by part status and for an open bid part; the part buyer can see the number of current bids. For finished parts, the buyer can see who the seller is and the price. A seller can see the parts on which he/she placed bids and the seller can also get a list of successful or unsuccessful bids.



My Bids/part requested

Here you can see all of your bidding parts as a seller or parts you offer as a buyer at a glance. You can view any part information by clicking the Part number. The current status of the part and your role determine what kind of information you can get.

New Part Offer

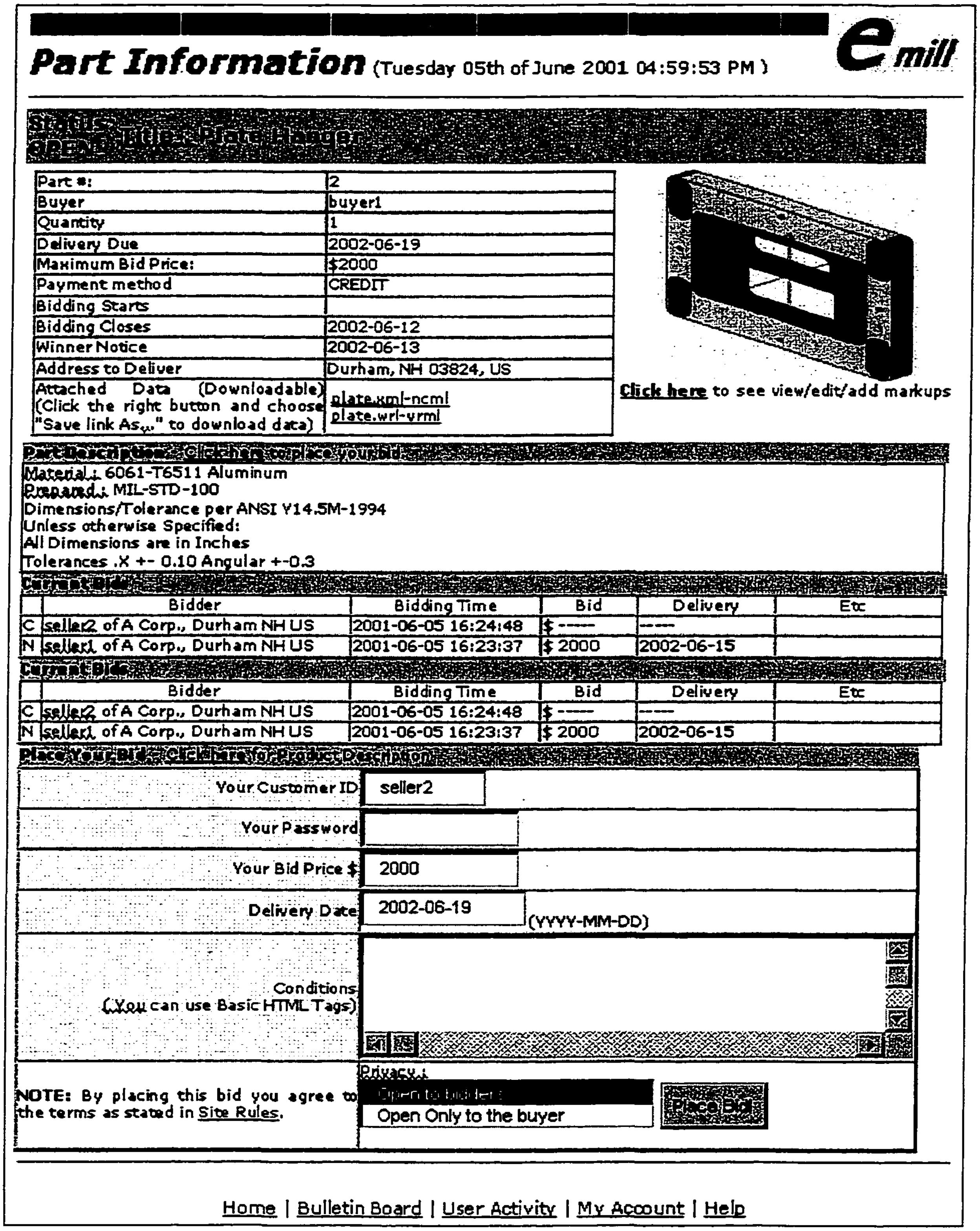
Select a winner for close parts

Open Parts

Parts being prepared to open

Finished Parts

Figure 5.11 E-Mill user activity page



Part Information (Tuesday 05th of June 2001 04:59:53 PM)

Status : OPEN Title : Plate Hanger

Part #:	2
Buyer	buyer1
Quantity	1
Delivery Due	2002-06-19
Maximum Bid Price:	$2000
Payment method	CREDIT
Bidding Starts	
Bidding Closes	2002-06-12
Winner Notice	2002-06-13
Address to Deliver	Durham, NH 03824, US
Attached Data (Downloadable) (Click the right button and choose "Save link As..." to download data)	plate.xml-ncml plate.wrl-vrml

Click here to see view/edit/add markups

Part Description : Click here to place your bid

Material : 6061-T6511 Aluminum
Prepared : MIL-STD-100
Dimensions/Tolerance per ANSI Y14.5M-1994
Unless otherwise Specified:
All Dimensions are in Inches
Tolerances .X +- 0.10 Angular +-0.3

Current Bids

	Bidder	Bidding Time	Bid	Delivery	Etc
C	seller2 of A Corp., Durham NH US	2001-06-05 16:24:48	$ -----	-----	
N	seller1 of A Corp., Durham NH US	2001-06-05 16:23:37	$ 2000	2002-06-15	

Current Bids

	Bidder	Bidding Time	Bid	Delivery	Etc
C	seller2 of A Corp., Durham NH US	2001-06-05 16:24:48	$ -----	-----	
N	seller1 of A Corp., Durham NH US	2001-06-05 16:23:37	$ 2000	2002-06-15	

Place Your Bid : Click here for Product Description

Your Customer ID	seller2
Your Password	
Your Bid Price $	2000
Delivery Date	2002-06-19 (YYYY-MM-DD)
Conditions (You can use Basic HTML Tags)	
NOTE: By placing this bid you agree to the terms as stated in Site Rules.	Privacy : Open to bidders Open Only to the buyer

Home | Bulletin Board | User Activity | My Account | Help

Figure 5.12 Part information displayed in E-Mill

5.2.4 Part Information Posting

A registered designer in the E-Mill service can post a machined part in NCML format through the uploading files section of the E-Mill web site. The data posted mainly includes a NCML document and the VRML 3D graphical model. If needed, other supplementary data can be included. The E-Mill application will arrange the given data on the specified web page and inform the sellers that an RFQ is ready for bid.

All information about the part (the information submitted by the part buyer) is displayed as shown in Figure 5.12. Sellers and buyers can view the graphical image of a part and download the data files (NCML, VRML or other geometry in an IGES file).

Sellers (Job shops) can place bids on the part and the buyer can choose a bid winner for his/her offer. This page also changes its contents according to part status and part ownership.

5.2.5 Design Collaboration-Communication Channel

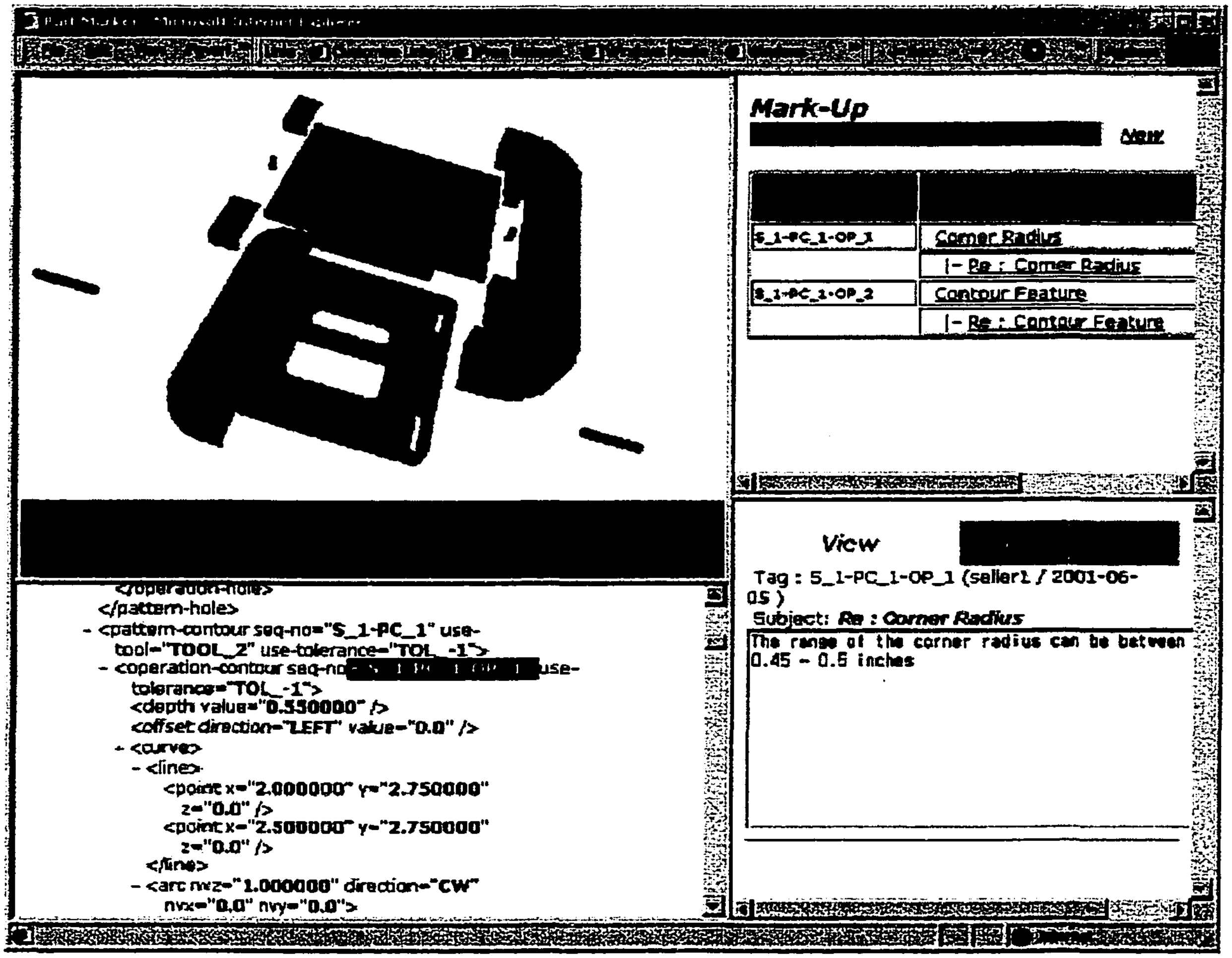
A seller in this scenario or a potential part bidder can examine the design information (geometry and specifications) and interact with the part buyer by using the markup capability. The left upper frame in Figure 5.13 shows a standard VRML viewer plug-in (Cosmo player). Users can rotate and zoom the image in and out. They can change the machining set up defined in the NCML document in the left lower frame. A user can pop out a fabrication feature by clicking on it in the VRML frame. This also leads the user to the geometric definition of the selected feature in the NCML frame.

Clicking a line in the right upper window brings up the markup on the right lower window. Sellers can create and add new comments on the part on the right side of this page. By responding to the sellers' comments, the buyer can resolve ambiguities in part design.

5.2.6 Match Making Mechanism

As mentioned earlier, E-Mill will be a B2B e-commerce web site that connects buyers and sellers of custom machined parts. The most successful and effective matchmaking scheme on the web today is the auction or bidding scheme. We can find examples of this method in well-known sites like www.ebay.com or www.ubid.com, which also sell their experience and technology to others who want to adopt similar models.

The match-making model applied in E-Mill is a variant of the bidding model used in other well-known auction sites. The main difference is that price is of primary importance when buying and selling commodity items, e.g. an HP printer or a Sony digital camera. In the E-Mill model, delivery time and the buyer's confidence in the Job Shop's capabilities are at least as important as price. E-Mill will match buyers and sellers efficiently by allowing buyers to set the bidding period, specify a target price and/or delivery time, and by allowing the sellers to suggest modifications in a convenient electronic format.



```
      </operation-hole>
    </pattern-hole>
  - <pattern-contour seq-no="S_1-PC_1" use-
      tool="TOOL_2" use-tolerance="TOL_-1">
    - <operation-contour seq-no[redacted]use-
        tolerance="TOL_-1">
        <depth value="0.550000" />
        <offset direction="LEFT" value="0.0" />
      - <curve>
        - <line>
            <point x="2.000000" y="2.750000"
              z="0.0" />
            <point x="2.500000" y="2.750000"
              z="0.0" />
          </line>
        - <arc nvz="1.000000" direction="CW"
            nvx="0.0" nvy="0.0">
```

Mark-Up

New

S_1-PC_1-OP_1	Corner Radius
	|- Re : Corner Radius
S_1-PC_1-OP_2	Contour Feature
	|- Re : Contour Feature

View

Tag : S_1-PC_1-OP_1 (seller1 / 2001-06-05)

Subject: ***Re : Corner Radius***

The range of the corner radius can be between 0.45 - 0.6 inches

Figure 5.13 A screenshot illustrating the design/fabrication collaboration using VRML graphical NCML display (Top-left window) and markups (right windows)

5.3 FACILE/Fabricate

The NCML file, which may be downloaded from the E-Mill service, can be imported to FACILE/Fabricate. The imported NCML document is translated into an internal solid model and viewed by the seller. By adding machining specific information to the delivered NCML document, the fabricator can implement the whole fabrication process in the FACILE/Fabricate system. This includes cost estimation for quote generation and G-Code generation.

5.3.1 Overall Procedure and Data Structures

In order to generate toolpaths in FACILE/Fabricate, it is necessary to supply some additional information to the NCML file. This fabrication centered data is used to describe the fabricator's machining capabilities. The required data includes:

- Machine Tool library: machine tools are characterized and stored in this data structure. For each machine, specific machine attributes are determined by characterizing physical properties, like work piece size, or attainable tolerance level.
- Machining (Cutting) Conditions library: Machining conditions, like feed rate or spindle speed, that are specific to each machining operation are parameterized and kept in this library. There are also a number of parameters that are operation specific, for example depth of cut for each pass in a pocket or the amount of material left on a contour for the finishing cut. Users can define multiple sets of machining conditions and should choose a set that is appropriate for the specific part to be machined. This machining condition set is used to generate G-codes from the NCML.
- Quote Parameter library: To support the quotation helper in FACILE/Fabricate, some parameters used with the quoting process can be formed as a set of parameters and stored in this library. Material removal rates related to machining operations, variants of the weighting factor according to tolerance values, and cost information are kept in this library.
- Cutting Tool library: Cutting tool information should be referred to when assigning appropriate tools to the machining operations of the process plan. Machining conditions will be changed according to the tool assigned to the operation. Cutting tool geometry information is also used in generating G-codes to

avoid interference with the work piece. The available tools are stored in this file and presented at the process planning stage.

- Tool Turret library: Cutting tools used in process planning should be assigned a unique tool turret index number. This tool turret information can be created temporarily and be stored in the Library for future use.

FACILE/Fabricate generates quotes and G-codes. To enhance the reader's understanding, the basic data structures used in implementing FACILE/Fabricate are provided in Figure 5.14 with the associated processes.

The NCML may be checked to see if the part can be machined with the machine described in "Machine Tool". NCML workpiece size and tolerance information are compared with those in the "Machine Tool" Library. If the NCML information falls into the possible machining range of the "Machine Tool" Library, this part is passed to the next process.

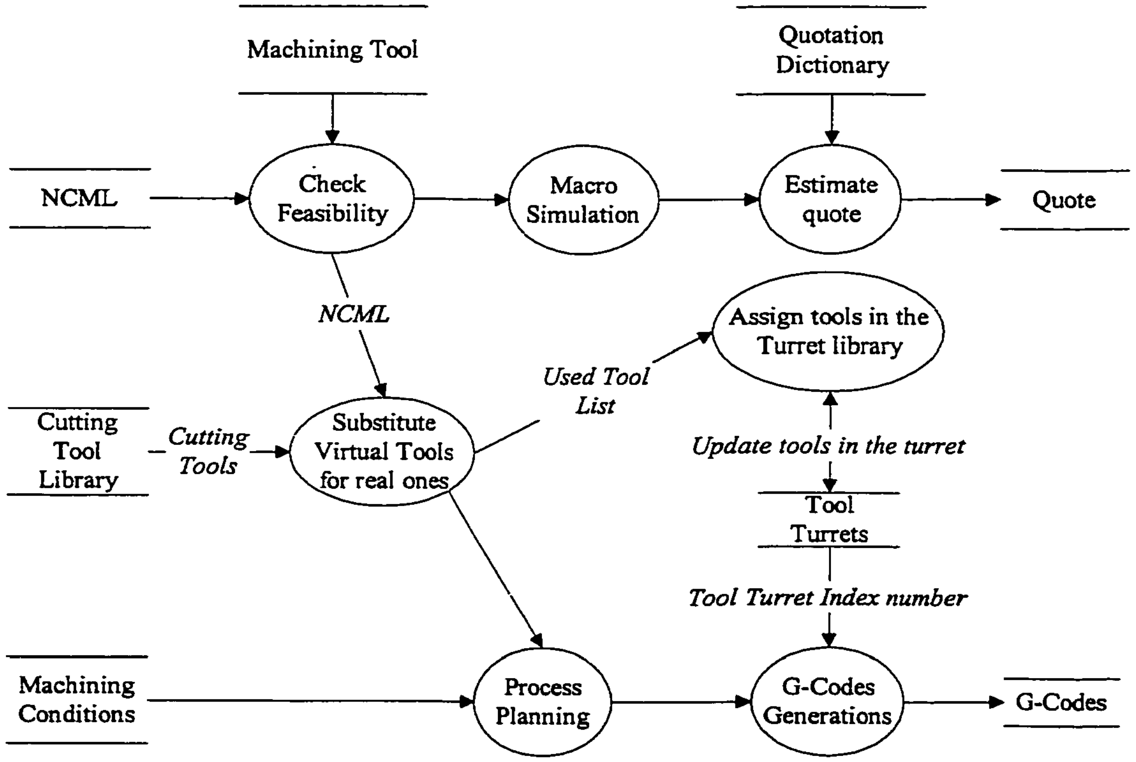


Figure 5.14 FACILE/Fabricate processes and data structures

The actual available tools from the "Cutting Tool" library should replace virtual tools. These actual tools should be assigned a tool turret number. The modified NCML and tool information will go through process planning and G-code generation.

The functions for quote estimation, process planning, and G-code generation are explained below. Detailed descriptions of the libraries dealt with here are given in Appendix C.

5.3.2 Quotation Helper-Quoting in FACILE/Fabricate

Chang [CHANG90-p135] mentioned the importance and difficulties of estimating quotes as follows:

> "It is always of interest to us to find the most economical solution. Often, it means the survival of the company. Basically, process economics means the cost efficiency of the processes. For mass production, it is necessary to go through a very detailed economic analysis before selecting a specific processing method. However, for the usual small to medium batch production it is not practical to conduct a very detailed study. The amount of time spent cannot justify the saving. Some rough estimation or just common sense should be used to select the best process...."

We assume that E-Mill employs an auction/bid method for matchmaking. Timely and accurate quotation is helpful for both buyers and sellers. Generating a quick and reasonable cost estimation of machined parts is crucial to competitive business in the machined part industry.

There are some cost estimation systems commercially available, such as the Machine Shop Estimating system (MSE)[1] from Micro Estimating Systems, Inc. MSE is an engineering-based process planning and cost estimating system for manufacturers. MSE calculates precise machining times and accurate product costs based upon company-specific equipment and estimating procedures. While it seems to provide an extensive array of tools for managing the entire estimating process and for creating accurate estimates, this system still heavily relies on a human intensive work process in recognizing machining features.

Here, a heuristic method for quote estimation of the machined part in NCML is presented. The proposed method fully utilizes information in the NCML format to generate a reasonable quote for the machined part in NCML.

Quotation Model

A quotation model[2] can be stated as:

$$C=C_m (T_m + T_p)+ Q (C_w + C_p) \tag{5.1}$$

where

C : Total cost for fabricating parts

C_m: Operator rate and machine overhead (\$/hr)

C_w : Cost of workpiece

C_p : Cost of part post-machining operations

T_m : Machining time for all quantities

T_p : Pre-machining processing Time

Q: Quantity of part

Pre-machining processing time (T_p) includes programming time for generating G-codes and documentation process time in general. Besides T_m, other components can be determined by experience. Machining Time (T_m) can be expressed as:

$$T_m=T_s+ T_c + T_t + T_u \tag{5.2}$$

where

T_s: Total setup time

T_c : Total cutting time

T_t : Total tool change time

T_u : Total work piece setup (load/unload time)

The time required for machining setup includes fixture making, cutting tool loading, trial

---

[1] Some information is available on the company web site (www.microest.com).

[2] This cost model is based on the quoting practice of the Stone Machine Company (www.stonemachine.com), Chester, New Hampshire, USA.

machining and machined part inspection. Since, the number of setups can be easily gained from the NCML format, T_s can be stated as:

$$T_s = N_s A_s \tag{5.3}$$

where

N_s : Number of setups

A_s : Average machining setup time

Since the number of cutting tool changes depends on the detailed process plan, it is quite difficult to get an exact number. However, since NCML associates a "virtual tool" with each individual pattern element, an approximation can be made. In this case, the total time required for changing cutting tools can be stated as:

$$T_t = Q A_t \Sigma N_t(i) / N_b \tag{5.4}$$

where

$N_t(i)$: Cutting tool change number for setup i, $i=1..N_s$

N_b = Batch size

A_t = Average cutting tool change time

Loading /unloading can be expressed as:

$$T_u = Q N_s A_u / N_b \tag{5.5}$$

where

$N_t(i)$: cutting tool change number for setup i, $i=1..N_s$

A_u : Average workpiece load/unload time

During Macro Simulation, machining volume and the area of each machining feature can be achieved by subtracting the post-operation workpiece volume from the pre-operation one. NCML cost estimation uses this material removal volume and machined surface area as the primary input to its algorithms.

Machined parts are usually machined to final shapes by a series of roughing and finishing paths with different-size tools. After rough cutting, in which a large cutter removes most material at large material removal rates, finish cutting is performed to smooth the rough surface. The cutter that affects the part final shape and desired tolerance does this finishing cut. Therefore, the calculated machined volume during the macro simulation is the material removed by both the rough- and finish-cutting operations. Though the time for rough cutting can be calculated by dividing machining volume by the specific material removal rate, the finish cutting time is proportional to the area of machined surfaces. Dividing the machined surface area by the surface cutting rate is used to estimate finish cutting time. Machining operations defined in NCML are of type HOLE, CONTOUR, POCKET, FACING, AND BOSS. The machined volume created from the CONTOUR or POCKET falls into the aforementioned roughing and finishing cutting category. The HOLE operation can be achieved by several individual operations; center drilling, drilling, reaming, or tapping. Each operation usually uses a different cutting tool. At this point, the cost estimator only regards the hole operation as a volume based cutting operation like the rough cutting in pockets or contours.

The FACING and BOSS operations are a volume removing operation like the pocket operation. The FACE cutter usually sweeps across a plane area, and the machining time depends on the machining area not the volume. Considering these discussions, the total cutting time can be stated as:

$$T_c = Q\,\Sigma(c_1\,V\,R_r + c_2\,V\,R_d + c_3\,A_w\,R_w + c_4\,A_f\,R_f),$$

$$\text{If operation type is hole, } c_1 = c_4 = 0,\ c_2 = c_3 = 1$$

$$\text{If operation type is contour, pocket or boss, } c_2 = c_4 = 0,\ c_1 = c_3 = 1$$

$$\text{If operation type is facing, } c_1 = c_2 = c_4 = 0,\ c_3 = 1 \qquad (5.6)$$

where

V: Volume removed from the current cutting operation

A_w : Area of wall created from the current cutting operation

A_f : Area of bottom surface created from the current cutting operation

R_r : Material removal rate for rough cutting

R_d : Material removal rate for hole machining

R_f : Bottom surface cutting rate

R_w : Wall surface cutting rate

c_1 ,.., c_4 : Permutation parameters

Although, in general, the cutting tool with a large diameter has a large associated material removal rate, it is difficult to derive an accurate relationship between material removal rate and the tool used. We assume the material removal rate is inversely related to the cutter diameter. Surface finish level and tolerance also affect the material removal rate. Material removal rates are stated as:

$$R_d = F_t R_{dr} (D_c/D_r)^3 \tag{5.7}$$

$$R_f = F_t R_{fr} (D_c/D_r)^2 \tag{5.8}$$

$$R_w = F_t R_{wr} (D_c/D_r) \tag{5.9}$$

where

R_{dr} : Reference MRR for hole machining

R_{fr} : Reference bottom surface cutting rate

R_{sr} : Reference wall surface cutting rate

D_c : Tool diameter of the current cutting tool

D_r : Tool diameter of the reference cutting tool.

F_t : Weight factor, changes with tolerance and surface finish assigned on the feature

Different power numbers are used in Equations 5.7-9 to adjust the material removal rate for specific operations. This value is determined mainly depending on which geometric element is machined. For example, power 3 is used for a 3 dimensional element volume, power 2 is used for a 2 dimensional element surface, and we assume that a cutter for wall machining moves in one dimension along lines and arcs, therefore power 1 is chosen. Although, this assumption may not be true for some sophisticated tool shapes and machining geometry, it predicts well in experiments.

General Procedure

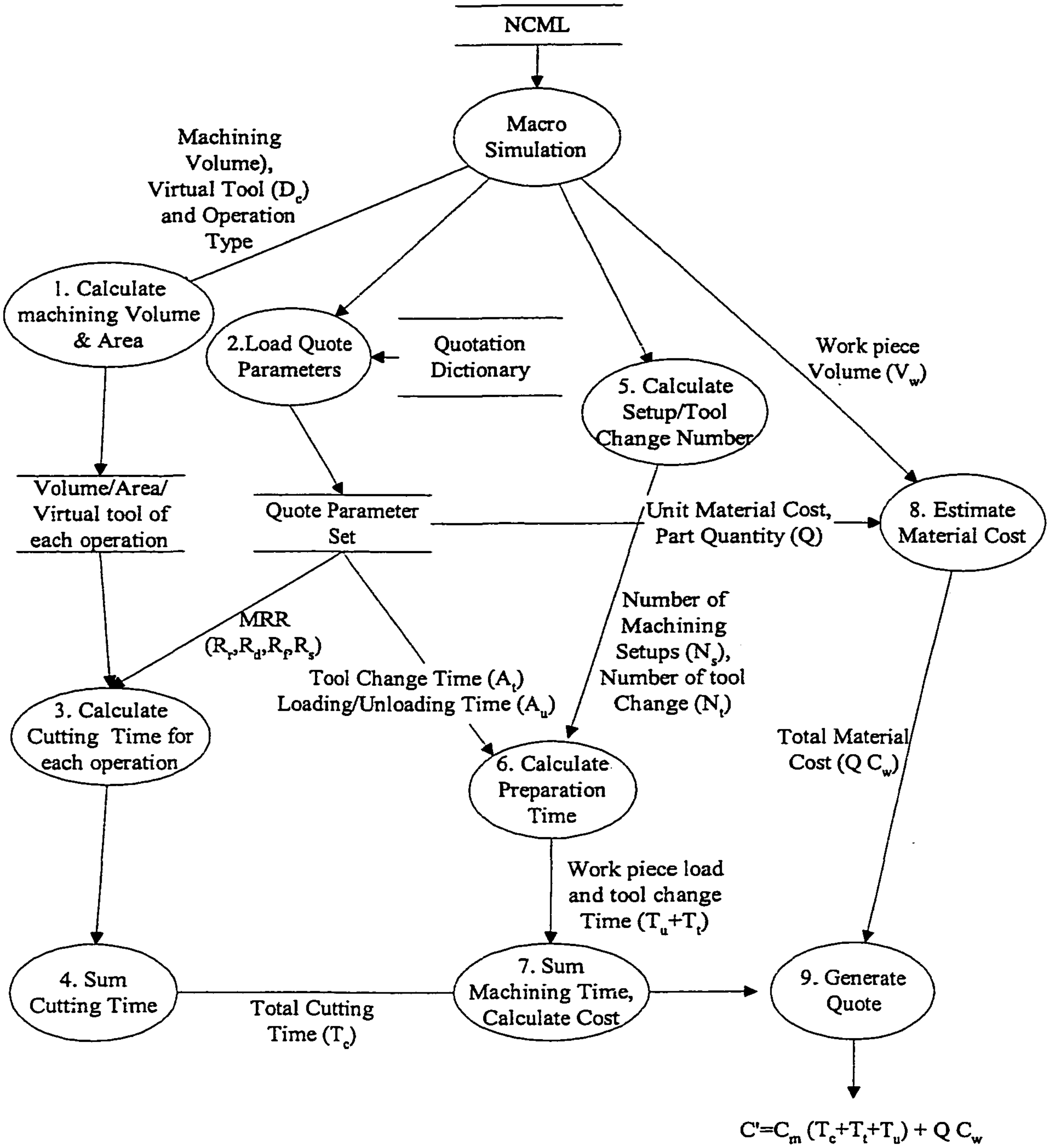


Figure 5.15 Overall procedure for NCML-based cost estimation in FACILE/Fabricate

The part fabrication cost is assumed to be the sum of the machining cost and the raw material workpiece cost. The machining cost can be divided into the cost directly related to cutting and the cost related to the tasks of material loading/unloading and tool change. Figure 5.15 shows a diagram explaining meaningful steps to create a machined part quote in FACILE/Fabricate. The numbers in the diagram in Figure 5.15 are equivalent to the following step number. The symbols in the cost model also are imposed on Figure 5.15.

Step 1. Macro simulation gives the volume for each NCML operation in addition to the operation type and description of the virtual tool. To calculate the cutting time for the finishing cut, the area of the cutting surface is necessary for pocket, boss, facing and contour operations. During machining operations some surface patches may disappear and others will be created. Clarifying which surface area is generated by the current operation is a complex problem. Averaging the machined volume by constant machining depth leads to the area of the machined surface. The algorithm for calculating the created surface area is given in Algorithm 5.1.

Step 2. The user can choose suitable quote parameters sets depending on the machined part characteristics.

Step 3. Calculate cutting time. This step is the process of calculating cutting time from the given cutting volume or area based on the MRR values. Various MRR values are gained from the quote parameter set loaded in Step 2. Volume based cutting time can be calculated by multiplying the material removal rate for rough cutting of POCKET and CONTOUR operations or material removal rate for drilling of the HOLE operation. The same method is used to calculate the finish cutting time. However, different MRR values are chosen according to the specified surface tolerance.

Step 4. The sum of all cutting time occurring in each NCML operation is the total cutting time.

Step 5. Extract the number of machine setups and possible number of tool changes. In NCML, the number of the machining setups is explicitly shown as a setup element. The time required to change cutting tools is relatively short so it may be ignored. However, tool change time is considered as a factor to affect total fabricating cost and an approximation in Algorithm 5.2 is applied to get the tool change count.

Step 6. Apply the average setup time and average tool change time to the number of setups and tool changes in Step 5 gives an additional time factor in the machining operation.

Step 7. Adding the results of Step 4 and Step 6 gives the total estimated time for machining. Multiplying the time estimates by the machining cost rate generates the machining cost.

Step 8. Material cost. Since NCML includes workpiece information, it is simple to get the material cost.

Step 9. Adding the operational cost in Step 7 to the material cost in Step 8 generates the cost estimation based on the NCML.

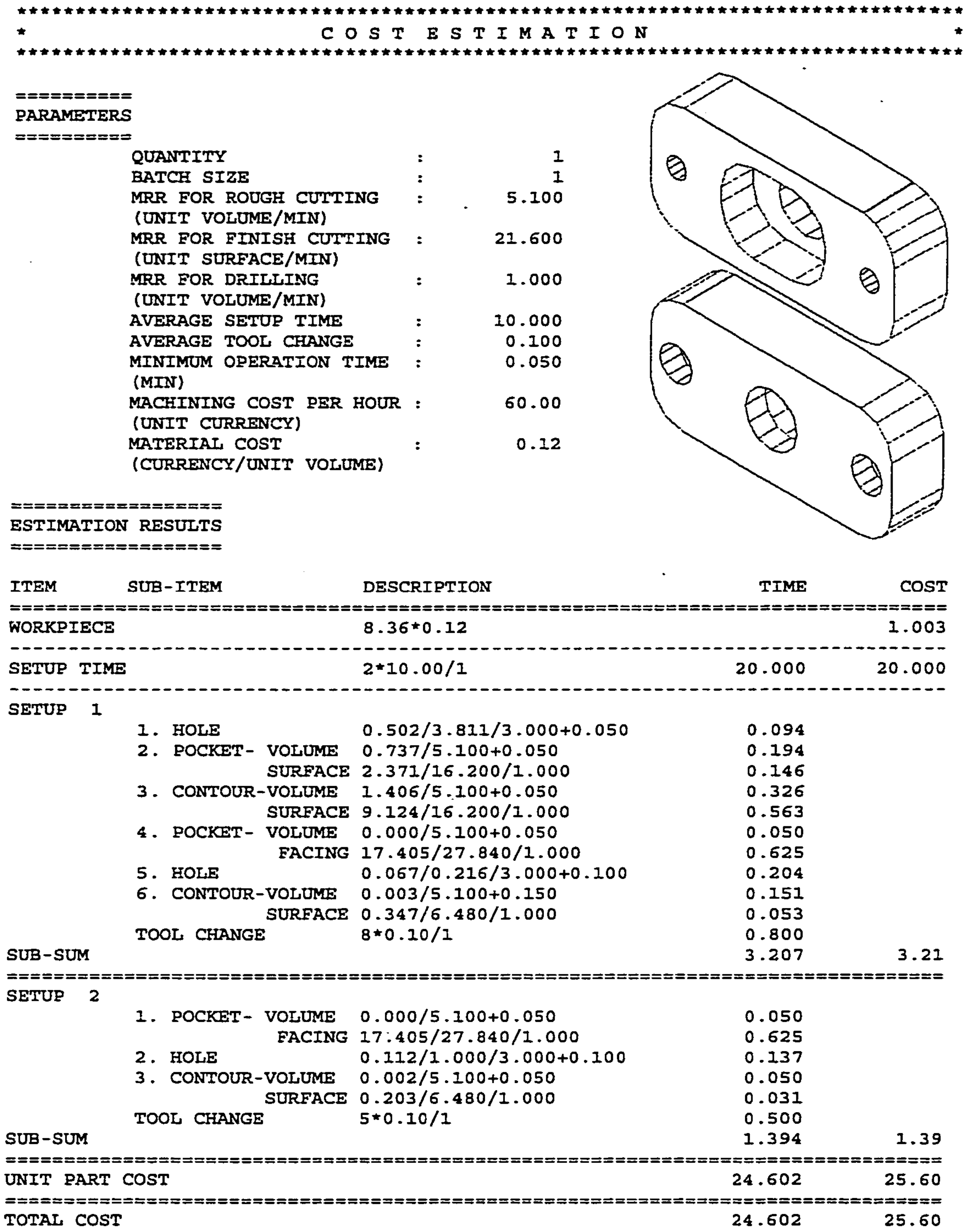
The cost estimation approach uses the machining volume and area of features to generate quotes. Experimental data such as programming time, machining setup, and fixture making may need to be added on the calculated quote from FACILE to get an estimation for the cost model provided. In the future, the Intelligent Machining Workstation can gather this data automatically and provide continuously improved parameters for quotation estimations.

Algorithm 5.1 Surface area calculation for finish cutting

```
machined_surface_area=0
IF NCML operation type is pocket
        machining_depth=machined_volume/pocket_area
ELSE IF operation type is contour
        machining_depth=machined_volume/(contour_length *cutting_tool_radius)
        // Assuming half tool immersion for any contour operation
If side wall surface finish cutting is needed,
        machined_surface_area += constant_height * pocket_periphery
If bottom surface finish cutting is needed,
        IF NCML operation type is pocket
                machined_surface_area += pocket_area
        ELSE IF operation type is Contour
                machined_surface_area += constant_height * contour_length
End of Algorithm 5.1
```

Algorithm 5.2 Tool change number acquisition

```
tool_change_no=0
for each machining setup,
        get number_pattern_hole, number_pattern_contour and number_pattern_pocket.
        tool_change_no+=( number_pattern_hole+number_pattern_contour + number_pattern_pocket)
                // Total number of patterns in the NCML means the number of different tools.
        if number_pattern_hole>0
                tool_change_no+=1          // Add center drill for each setup
        if (number_pattern_contour + number_pattern_pocket) >0
                tool_change_no+=1               // Add roughing tool for each setup.
End of Algorithm 5.2
```



```
*********************************************************************************
*                         C O S T   E S T I M A T I O N                         *
*********************************************************************************

==========
PARAMETERS
==========
          QUANTITY                 :           1
          BATCH SIZE               :           1
          MRR FOR ROUGH CUTTING    :       5.100
          (UNIT VOLUME/MIN)
          MRR FOR FINISH CUTTING   :      21.600
          (UNIT SURFACE/MIN)
          MRR FOR DRILLING         :       1.000
          (UNIT VOLUME/MIN)
          AVERAGE SETUP TIME       :      10.000
          AVERAGE TOOL CHANGE      :       0.100
          MINIMUM OPERATION TIME   :       0.050
          (MIN)
          MACHINING COST PER HOUR  :       60.00
          (UNIT CURRENCY)
          MATERIAL COST            :        0.12
          (CURRENCY/UNIT VOLUME)

==================
ESTIMATION RESULTS
==================

ITEM      SUB-ITEM            DESCRIPTION                       TIME        COST
================================================================================
WORKPIECE                     8.36*0.12                                    1.003
--------------------------------------------------------------------------------
SETUP TIME                    2*10.00/1                       20.000      20.000
--------------------------------------------------------------------------------
SETUP  1
           1. HOLE            0.502/3.811/3.000+0.050          0.094
           2. POCKET- VOLUME  0.737/5.100+0.050                0.194
                      SURFACE 2.371/16.200/1.000               0.146
           3. CONTOUR-VOLUME  1.406/5.100+0.050                0.326
                      SURFACE 9.124/16.200/1.000               0.563
           4. POCKET- VOLUME  0.000/5.100+0.050                0.050
                       FACING 17.405/27.840/1.000              0.625
           5. HOLE            0.067/0.216/3.000+0.100          0.204
           6. CONTOUR-VOLUME  0.003/5.100+0.150                0.151
                      SURFACE 0.347/6.480/1.000                0.053
           TOOL CHANGE        8*0.10/1                         0.800
SUB-SUM                                                        3.207        3.21
================================================================================
SETUP  2
           1. POCKET- VOLUME  0.000/5.100+0.050                0.050
                       FACING 17.405/27.840/1.000              0.625
           2. HOLE            0.112/1.000/3.000+0.100          0.137
           3. CONTOUR-VOLUME  0.002/5.100+0.050                0.050
                      SURFACE 0.203/6.480/1.000                0.031
           TOOL CHANGE        5*0.10/1                         0.500
SUB-SUM                                                        1.394        1.39
================================================================================
UNIT PART COST                                                24.602       25.60
================================================================================
TOTAL COST                                                    24.602       25.60
```

Figure 5.16 Cost estimation result sheet

<u>Example</u>

Figure 5.16 shows a sample page of the cost estimation results. On the right side of Fig 5.16, there

are two isometric views showing the front and the rear sides of a sample part that has several manufacturing features. It was directly imported to FACILE in the form of a NCML document. The result page is divided into two parts: one describes the user-input parameters and the other is for the estimation itself. The estimation part shows the operation time and cost, which are categorized by fabrication processes: material cost, machining setup, tool change, cutting operation, etc. In this example, the result shows the quote for the part below is $25.60. At first glance, this might seem like a poor estimate since Stone Machine Co. charges their customer around $15 for a very similar part. However, the material costs plus the machining cost for the two setups totals $5.60 ($1.00 + $3.21 + $1.39). When made in quantity, this represents a lower bound on the part cost which would be very difficult to decrease. Through clever fixturing and volume production, the setup times can become almost negligible. The results clearly indicate the effect that part quantity has on cost.

5.3.3 Process Planning

Chang [CHANG90] states the process plan as a recipe: "In a sense, a process plan can be considered a recipe. A recipe is 'a set of instructions for making something from various ingredients'. Given a detailed recipe, a novice cook can usually prepare almost any meal".

Chang defines process planning as an act of preparing detailed processing documentation for the manufacture of a part or assembly. The purpose of the process planner of FACILE/Fabricate is to carry out prerequisite steps for generating instructions (G-codes) for the automated machines (NC machines). In other words, any activities required for transferring NCML information into G-Codes are treated as process planning activities in FACILE/Fabricate. The overall procedures for process planning are shown in Figure 5.14. More detailed information on the FACILE/Fabricate software functions is given below.

Translation from NCML

The NCML document must be translated from the machining features (NCML operations) into a process plan consisting of a series of completely specified Unit Machining Operations (UMOs)[CJ98]. The UMOs must include all of the required process information, including depth of cut, roughing and finishing operations, pocketing strategies [HGS98] (i.e., spiral vs. zigzag), spindle speed and feedrates. Since the specific process plan depends heavily on the capabilities of the individual machine shop, this information is not included in NCML. Indeed, it is one of the governing philosophies of NCML that only the information essential to defining the goal be provided and the individual details be left to the cleverness of the fabricator.

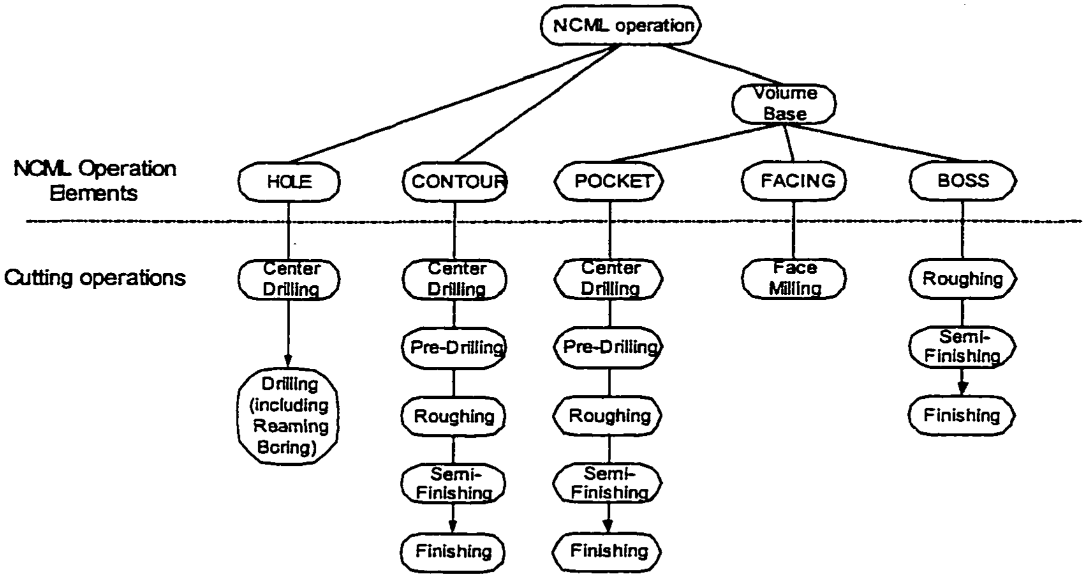
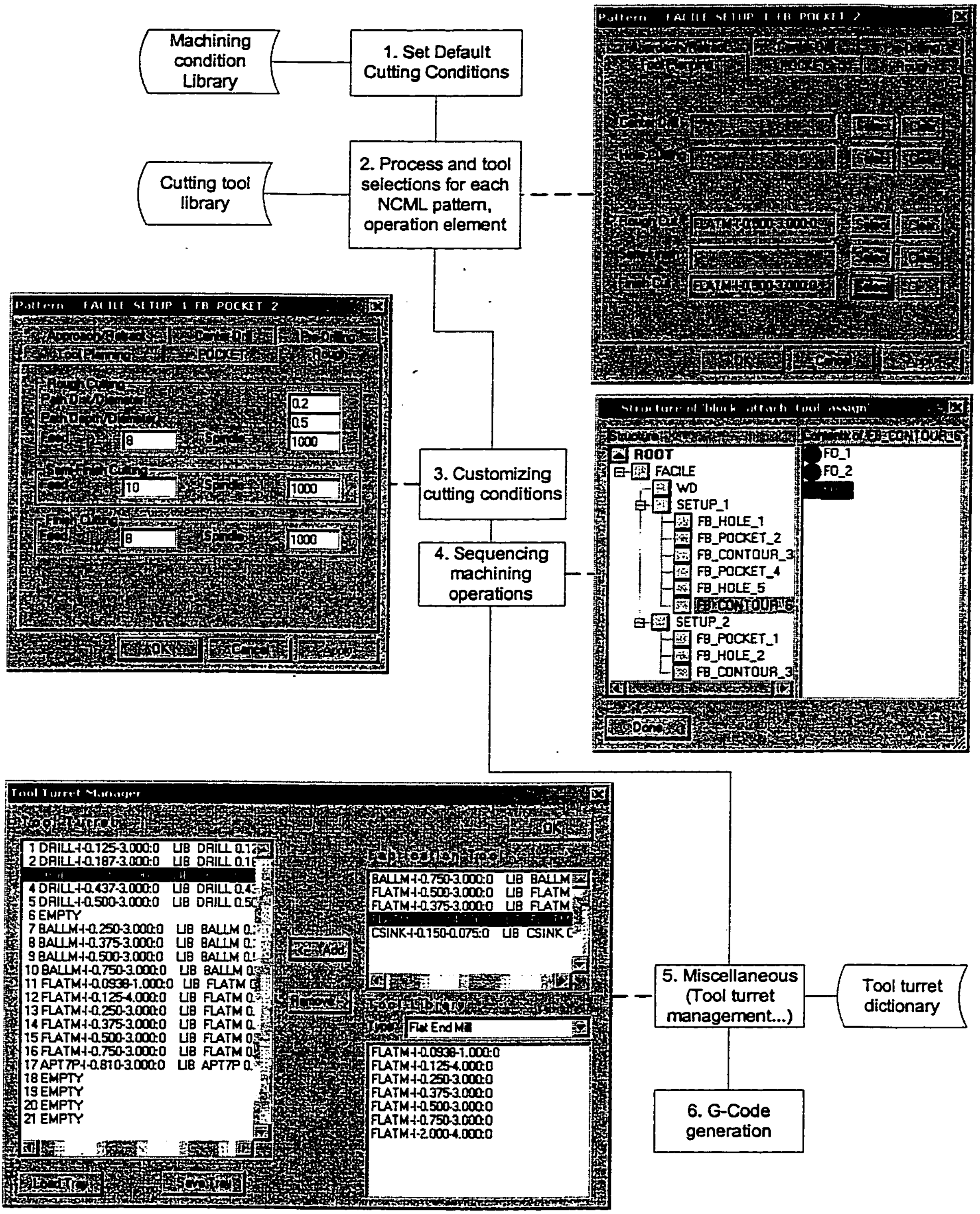


Figure 5.17 Decomposing of NCML operation elements to cutting operations

The goal of this process is to determine exactly how to machine the NCML operation elements: HOLE, CONTOUR, POCKET, FACING, and BOSS. As seen in Figure 5.17, it is assumed that each NCML operation can be represented as a set of UMOs. Based on this assumption, the next step is to assign cutting tools and cutting conditions to each NCML operation and its associated UMOs. Since each NCML operation does not always need to use all of the cutting operations in its category, the fabricator can select a subset of the UMOs dependent on the machining requirements. For example, an NCML POCKET operation can be machined by the five UMOs shown in Figure 5.17. The first two UMOs (Center Drilling and Pre-Drilling) are implemented to provide a proper tool approach and the other three operations are machining UMOs which achieve the intended shape. However, fabricators may decide not to execute the semi-finishing or finishing operations if the raw material is easily machined and the fabricator can achieve the required surface quality with only rough cutting. Work piece material properties, tolerances and cutting tool choices all affect the fabricator's choices.



Rough Cutting: Path Dist/Diameter 0.2; Path Depth/Diameter 0.5; Feed 8; Spindle 1000
Semi-Finish Cutting: Feed 10; Spindle 1000
Finish Cutting: Feed 8; Spindle 1000

- ROOT
  - FACILE
    - WD
    - SETUP_1
      - FB_HOLE_1
      - FB_POCKET_2
      - FB_CONTOUR_3
      - FB_POCKET_4
      - FB_HOLE_5
      - FB_CONTOUR_6
    - SETUP_2
      - FB_POCKET_1
      - FB_HOLE_2
      - FB_CONTOUR_3

FO_1
FO_2

```
1 DRILL-I-0.125-3.000:0   LIB DRILL 0.12
2 DRILL-I-0.187-3.000:0   LIB DRILL 0.16
4 DRILL-I-0.437-3.000:0   LIB DRILL 0.43
5 DRILL-I-0.500-3.000:0   LIB DRILL 0.50
6 EMPTY
7 BALLM-I-0.250-3.000:0   LIB BALLM 0.
8 BALLM-I-0.375-3.000:0   LIB BALLM 0.
9 BALLM-I-0.500-3.000:0   LIB BALLM 0.
10 BALLM-I-0.750-3.000:0  LIB BALLM 0.
11 FLATM-I-0.0938-1.000:0 LIB FLATM 0.
12 FLATM-I-0.125-4.000:0  LIB FLATM 0.
13 FLATM-I-0.250-3.000:0  LIB FLATM 0.
14 FLATM-I-0.375-3.000:0  LIB FLATM 0.
15 FLATM-I-0.500-3.000:0  LIB FLATM 0.
16 FLATM-I-0.750-3.000:0  LIB FLATM 0.
17 APT7P-I-0.810-3.000:0  LIB APT7P 0.
18 EMPTY
19 EMPTY
20 EMPTY
21 EMPTY
```

```
BALLM-I-0.750-3.000:0  LIB BALLM
FLATM-I-0.500-3.000:0  LIB FLATM
FLATM-I-0.375-3.000:0  LIB FLATM
CSINK-I-0.150-0.075:0  LIB CSINK
```

Type: Flat End Mill

```
FLATM-I-0.0938-1.000:0
FLATM-I-0.125-4.000:0
FLATM-I-0.250-3.000:0
FLATM-I-0.375-3.000:0
FLATM-I-0.500-3.000:0
FLATM-I-0.750-3.000:0
FLATM-I-2.000-4.000:0
```

Figure 5.18 Process planning based on the NCML contexts in FACILE/Fabricate

After the machining operations from NCML operation elements are chosen and the cutting conditions for each machining operation are assigned, the next step in process planning is determination of the sequence of machining operations. Since NCML already includes the overall sequencing information in its internal structure, FACILE/Fabricate simply has to determine the order and properties of the UMOs required to implement each operation.

The process-planning activities are done in FACILE/Fabricate by following the process outlined in Figure 5.18. Some screen shots of the FACILE/Fabricate process planning are given to enhance the reader's understanding. The numbers shown in the blocks in figure 5.18 are identical to the numbers in the following procedural steps.

Step 1. Loading machining (cutting) conditions into FACILE: Users can upload the default cutting conditions applicable to all machining operation generated from the NCML. Cutting conditions include parameters like the feed rate, and the spindle speed applicable to each machining operation category. The loaded cutting conditions are assigned at the highest node (*setup*) in the NCML hierarchical structure and inherited by the lower nodes (through *pattern* to *operation* inheritance).

Step 2. Selection of cutting operations for the NCML pattern node: Users can assign an actual cutting tool for the cutting operations of each NCML pattern type. This activity includes substituting the virtual tool in the NCML pattern node by an actual cutting tool. In Figure 5.18, for a pocket pattern, a flat end cutter (indicated as FLATM-I-0.500-3.000:0) is assigned in both roughing and finishing operations. This means all pocket operation under this pattern element will be machined through roughing and finishing with a flat-end milling cutter of diameter 0.5.

Step 3. Customizing cutting conditions according to each operation characteristic: Information inheritance along the hierarchical structure is an important characteristic of the FACILE/Fabricate process planning. This may reduce the user's labor of inputting redundant machining condition for each operation. However, FACILE does allow users to adjust the machining conditions. As seen in the screen shot attached to the box indexed 3 in Figure 5.18, users can apply different machining conditions for each pattern or operation. For example, machining conditions are grouped into several categories (Tool planning, Pocket, Roughing, etc), and in the Rough category users can change the feed rate and spindle speed applicable to machining operations.

Step 4. Sequencing of machining operations: Changing the index number in the node name can easily change operation sequence.

Step 5. Assign cutting tools to be used in the tool turret: To generate G-codes, the turret location of each cutting tool used in machining must be known by the program. As seen in the figure, the tools to be used are shown in the right upper frame and users can assign these tools in the left frame, which is intended to model the tool turret. This tool turret information can be saved and recalled for future use.

The process planning in FACILE/Fabricate is automated, requiring a minimum level of the fabricator's effort. This process fully utilizes the fabrication related information like operation types, virtual tools, and operation sequence embedded in NCML and relies on cutting condition information imported from a saved library.

After carrying out the previous steps, we are now ready to generate machine instructions for the NC machines' G-codes automatically.

5.3.4 Tool-Path Generation

In the process plan, NCML operations are translated internally into UMOs for creating machine instructions. Required conditions for machining instructions are already assigned to each UMO at the process-planning phase. Thus, Tool-Path generation implies creating G-codes by referring to the geometric information and the cutting conditions of the UMOs.

As seen in Figure 5.17, FACILE defines 6 different UMOs - Center Drilling, Drilling (Pre-Drilling), Pocket-Roughing, Contour-Roughing, Semi-Finishing and Finishing, according to the geometric characteristic used in creating the tool paths. These UMOs can be placed into 3 different classes as follows:

- Hole making: Machining operations that can be defined by a point and a depth belong to this class of operations. The finished shape of the machining operation can vary with the cutter shape. Center

Drilling and Drilling (Pre-Drilling) belong to this operation class

- Contouring: Machining operation that can be defined by one or more curves and cutting depths are contours. The cutter moves along the curve or offset from the curve by the cutter radius. Semi-Finishing, Finishing and Roughing for CONTOUR belong to this class of operations.
- Pocketing: Cutting out a volume from a bounded area is usually called pocketing. The UMO of Roughing for Pocket operation belongs to this class.

It is straightforward to create the appropriate tool path for Hole Making and Contouring automatically. For example in Hole-Making the hole center and its depth are directly used for defining the tool path. By some tool position adjustments like offsetting or z-level change to accommodate cutting depth, tool path generation for contouring can be done very simply.

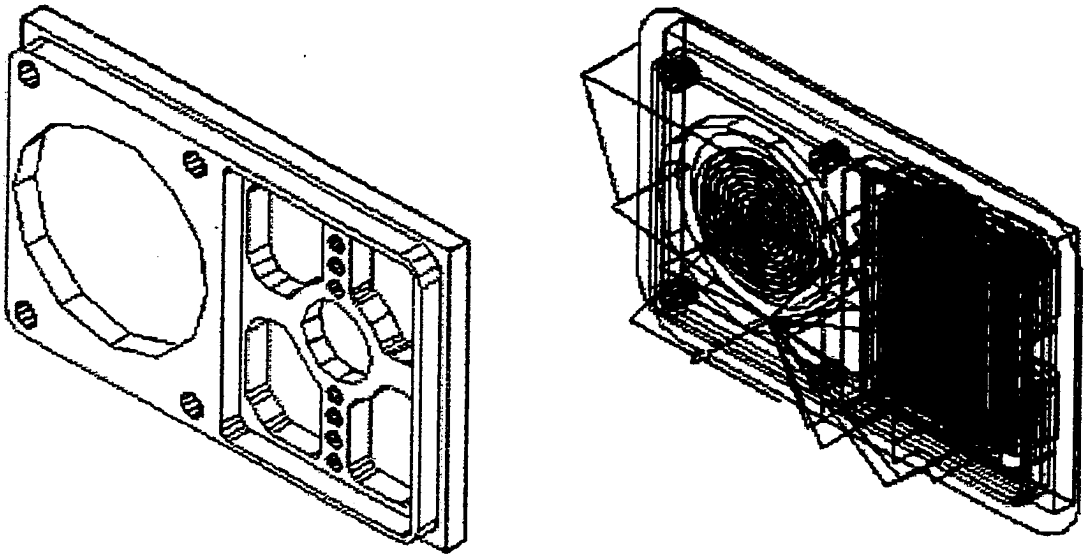
While the tool path for the pocketing of simple boundary geometry can be achieved by simply offsetting boundary curves, in real cases the complex boundary geometry makes it impossible to implement a robust pocket machining algorithm in the available time. Currently, FACILE/Fabricate can only provide tool paths for pocketing whose boundary is convex and does not contain any island geometry. Although it would have been nice to have such capability, it is not really a research issue since many commercial CAM software systems like "MasterCAM"[3][MC20] or "FeatureCAM[4]"[FS20] already have robust solutions for pocketing operations. Held et al. [HELD91, HLA94] has studied solving the problems associated with pocketing and readers are referred to these works.

Figure 5.19 shows the example of tool paths generated from FACILE. For illustration purposes, the tool path is imposed on the finished part geometry.

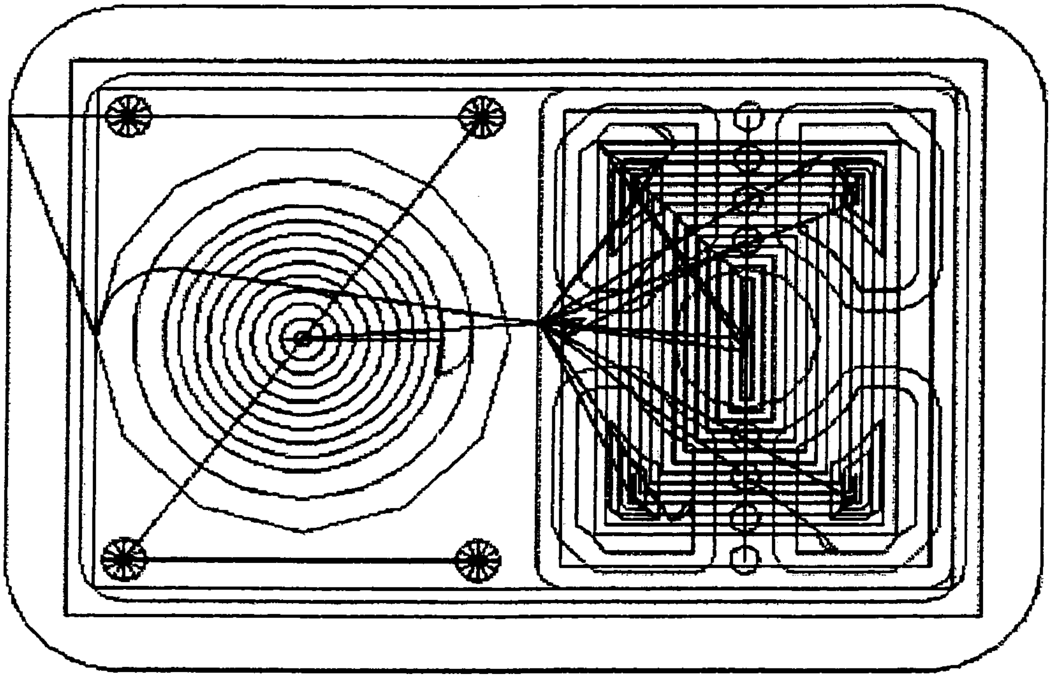
---

[3] One of the most widely used multi-purpose CAM systems from CNC Software, Inc (www.mastercam.com).

[4] FeatureCam from Engineering Geometry Systems is able to compile fully defined UMOs from a set of user defined machining features (www.featurecam.com)



(a) Machined part with tool paths in the isometric view



(b) Tool paths seen from top view

*Conventions: blue lines or arcs-G0, red -G1, G2, G3

Figure 5.19 An example of generated tool paths in FACILE

CHAPTER 6

SYSTEM TESTING METHODOLOGY

The previous three chapters have described the NCML language and the associated software system for fabricating machined parts. The proof of the concept can only be tested by applying it to the fabrication of actual parts and by comparing it to the conventional approach. This chapter describes the required tests and procedures for validating the NCML methodology.

6.1 What Can Be Tested?

The main purpose of the NCML process is fabrication of machined parts. As long as this functional requirement is met, the process is considered to be successful. However, NCML also introduces some additional functional capabilities that are not available in the conventional process. Measuring the benefits of these functional capabilities is an important part of the test.

The validity of the method lies in the answers to the following general questions:

1. Does the process allow the designer and fabricator to achieve the designed part?
    a. Is there any environmental barrier to introducing the new process?
    b. Is NCML general enough to represent general shapes of the prismatic part domain?
    c. Is the machined part fabricated efficiently with the NCML process?
2. Does the additional functional capability outweigh any inconveniences?
    a. Is the NCML document easy to create?
    b. What are the advantages of the additional functional capabilities?
    c. How useful are the additional functional capabilities?
    d. Is there any practical limitation to applying the additional functional capability to real situations?

The answer to the question, "Does the process allow the designer and fabricator to achieve the designed part?" can only be obtained by implementing the proposed design-and manufacturing process and actually producing finished parts. This is the most basic task required for the process validity. The modeling limitations of the NCML representation should also be tested. The choice of a good set of valid test parts is a key issue in answering these questions.

The answer to the second question, "Does the additional functional capability outweigh any inconveniences?" is a bit more obscure. Certainly, there is a major inconvenience associated with creating the NCML document. However, if the NCML process were commonly used, the transformation of a part into NCML could be treated as a commonly required activity like creating standard EDI files - STEP, IGES, etc. It is imperative that the benefits of the additional functional capabilities be quantified since the adoption of NCML is not without costs, and these costs must be offset by the associated benefits. FACILE and E-Mill functions like Macro simulation, e-commerce, Quotation helper, and Quick Process Planning are examples of the special functional characteristics which are enabled by NCML and the potential benefits of those capabilities were presented in the previous chapters.

6.2 Test Designs and Benchmark Developments

To answer the questions of the previous section, the following three types of tests are considered:

- Base Requirement test: Fabricating parts through the NCML process. Specifically verify the process all the way from a NCML file to a finished machined part.
- Quotation Validation test: Evaluation of quotation helper
- NCML Modeling Capability test.

The first two tests are designed to answer the respective questions of the previous section, and the third one is essential to know how well common machining features can be expressed in the NCML format.

6.2.1 Part Creation Test

The NCML data must be transformed into a physical machined part. This is the ultimate goal of the NCML process, which must be validated. The following items need to be measured in this fabrication process:

- How much time is required to turn the NCML file into machining operation codes, i.e. G-codes?
- Is the final machined part an accurate rendering of the desired part?

Measuring the NCML processing time gives a general sense as to how efficient this process is compared to conventional fabrication operations. A set of test parts must be chosen which are representative of the range of types found in actual practice.

The procedures for this fabrication test are the same as those described in FACILE/Fabricate. Test parts are chosen from the following sources:

- Stone Machine Company in Derry, New Hampshire, USA
- MfgQuote.com (http://www.mfgquote.com)
- National Design Repository at Drexel University (http://repos.mcs.drexel.edu/)

Test parts, which can be fabricated at any normal job shop, are available from Stone Machine Company. Actual machining times and costs of these parts is also supplied. Test parts typical of those being used in e-commerce can be downloaded from MfgQuote.com. After the RFQ process is complete the bidding statistics (high/low/average/median) are available, providing a valuable benchmark for the NCML method. The National Design Repository at Drexel University is a digital library, which is used as a source of test parts for research problems, particularly for researchers working on advanced process planning systems. More complex, difficult to produce parts can be found there.

Although more confidence is gained as more parts are machined, time constraints limit the study to ten parts. The chosen parts are fabricated using the FADAL CNC milling machine in the Design and Manufacturing Lab at the University of New Hampshire.

6.2.2 Evaluation of Quotation Helper

The ability of a job shop to accurately quote a part is directly related to their business revenues. Therefore, the consistency and the accuracy of the quote are essential for profitability. Since the quotation process is highly correlated with the actual machining process, the bids will vary widely for the same object. Each job shop has its own unique capabilities that may allow them to be more competitive on certain types of operations. The quotation is directly related to the accumulated experience in machining operations and the fabricator's skill. A single inflexible quotation system would be useless and any quotation helper application must be adaptable to reflect the practices of each individual job shop. The advantage of the "Quotation Helper" developed in this research is that as long as the part information is in the NCML format; the processing cost is almost zero. Providing accurate and inexpensive quotations is a great advantage, particularly in the highly competitive situation likely to be found in e-commerce.

To validate quotation helper, cases from real job shops must be considered. Machining operations and associated activities need investigation in order to calibrate the parameters used in estimation. Subsequently, this calibrated information can be used to estimate a quote for the part fabricated by the particular job shop. The estimated quote can be compared to the real cost of the part. The estimate should satisfy the following conditions:

- The estimate should accurately reflect the actual cost of production
- The estimates of a particular shop should be consistent, despite wide variations in job shop operations and experience, i.e. the software should be capable of being customized for each shop.

To verify the applicability of the method to a typical job shop, we investigated the fabricating practice at the Stone Machine Company located in Chester, New Hampshire, USA. We calibrated function parameters and generated the part cost estimate and compared these to the actual values. The following estimation outputs were compared to experimental values in real job shop operation:

- Machining time for an individual part
- Quotation for the whole job, which often involves quantities greater than just one part.

To investigate the applicability of the methods to e-commerce, we applied Quotation helper for parts posted on MfgQuote.com, which provides a web based bidding process. The parameters from the Stone Machine Company were used to calibrate the quotation helper software for these jobs. Comparing the estimates based on the Stone Machine capabilities to the actual bids in a real business provided an estimation of the system accuracy.

6.2.3 NCML Modeling Capability Test

Even if NCML could be used to produce parts in a very competitive way, it would not be successful if it could only produce a limited range of parts. NCML is currently targeted toward 2 ½ dimensional prismatic parts with manufacturing features producible on conventional 3-axis machining centers. Many researchers have studied machining features and tried to classify shapes into defined form features. These shapes can provide a good test for NCML's modeling capability. NCML must be tested to see if it is capable of expressing a versatile set of machining features.

Many classifications of geometric Machining/Manufacturing features are found in [CHANG90, FKH96, KRMER92]. The classification developed by Feng, Kusiak and Huang used in reference [FKH96] as used as the benchmarks for this research.

6.3 Experimental Protocols for Quotation Helper

6.3.1 Calibrating Quotation Helper

For accurate estimation, the machining environment needs to be considered. Parameters related to the cutting operation like material removal rates can be determined by collecting machining time data. Others parameters, such as tool change time or setup time may be determined by experience. In the process-planning phase, feed rates and spindle speeds are input by the user and are reflected in G-codes. The feed rate and spindle speed variables can account for only part of the material removal rate. The required parameters may vary with the work piece material, cutting tool characteristics, or as a combination of both tool and work piece materials. If it is assumed that the job shop uses the same machining strategy for the parts belonging to the same category; same material, similar part size, the material removal rates would be useful for preparing quotations of the parts of the same category.

Machining operation data must be obtained by observing actual machining. The data acquisition and parameter calibration procedures are as follows:

1. Preparation
    a. Choose a test work piece material.
    b. Prepare a good timekeeping method.
2. Machining operation data acquisition.
    a. Find an appropriate machined part for the target material.
    b. Measure dimensions of the un-machined work piece.
    c. Measure work piece load/unload time.
    d. Measure cutting tool change time.
    e. Make a list of the cutting tools used.
    f. Measure the time between tool changes. One tool may be used for several UMOs. Try to record the time corresponding to a UMO. If it is difficult to isolate the UMO from the whole operation, discard the operation. Each separate machining time can be recorded as the UMO time. Record cutting tool information along with the cutting time.
    g. Determine the UMO type of the observed operation based on the FACILE UMO classification
3. Adding tolerance information to the measured operations
    a. From the drawings of the part, find the imposed tolerance information and surface finish level of each machining operation observed in the previous step
4. Machining volume and surface calculation

    There are two ways to get volume and area of the machining operations observed in Step 2.

    a. The user can calculate operation volume manually by referring to the drawing and work piece dimensions. Or,
    b. Create an NCML model in FACILE. Execute "Quotation Helper". Quotation report gives volume and area values for each NCML feature.
    c. Put the acquired values in the column of the corresponding machining operation.
    d. Each operation will be summarized as shown in one of the rows in the table of Figure D.1

(See Appendix D).

5. Normalization of material removal rate
    a. Calculate the material removal rate or the area cutting rate depending on the designated operation type and normalize these values by the standard cutter diameter. Material removal rates are assumed to be inversely related to the cutting tool diameter as explained in Chapter 5. Exact equations for the material removal rate relationship to the cutter diameter are given in section 5.3.2.
6. Repeat steps 2 to 5 until a large number of machining operations have been observed.
7. Data organization
    a. Sort data by the operation category, tolerance and surface finish level.
    b. For the same operation category, tolerance and surface finish, get the average values. Users can discard data that deviates too much from the average (outliers).
    c. Set the material removal rate of the most coarse tolerances as the base material removal rate
    d. Calculate the weighting factors that imply the change of the material removal rate according to different tolerance and surface finish.
    e. These steps need to be repeated for the material removal rate for finishing, drilling and facing in addition to roughing.
8. Set non-machining related parameters
    a. Most non-machining related parameters can be determined by experience. For example, unit material cost can be easily set using past part information.

CHAPTER 7

TEST RESULTS

7.1 Example Parts – Fabrication Test Results

Ten parts were chosen from three different design sources (three from the Stone Machine Company, three from Mfgquote.com and four from the National Design Repository). Since some parts from the Stone Machine Company and MfgQuote.com are covered by a nondisclosure policy, only the parts not covered by the policy are presented as examples. Four of the ten parts were made with T6-6061 Aluminum and a plastic material was used for the remaining six parts. Figures 7.1 to 7.4 graphically illustrate the transition of four of these parts as information is added in the NCML based process. Each example part in Figures 7.1-7.4 shows four different phases.

- The first image shows the FACILE/Design schematic drawing of the example part. From this FACILE/Design model, a NCML document of the part is generated.
- The second image is the VRML model of the part, which is generated from FACILE/Design with the NCML document. Each NCML operation is given as a machining volume. Clicking each volume will simulate the machining operation. This VRML model is used as a graphical communication aid to accompany the NCML document in E-Mill.
- The third image shows toolpaths generated by the process planning module of FACILE/Fabricate superimposed over the final part shape.
- The fourth and the fifth images show the actual machined part after executing toolpaths on the CNC machine. The fourth image is shown with a ruler for comparative purposes.

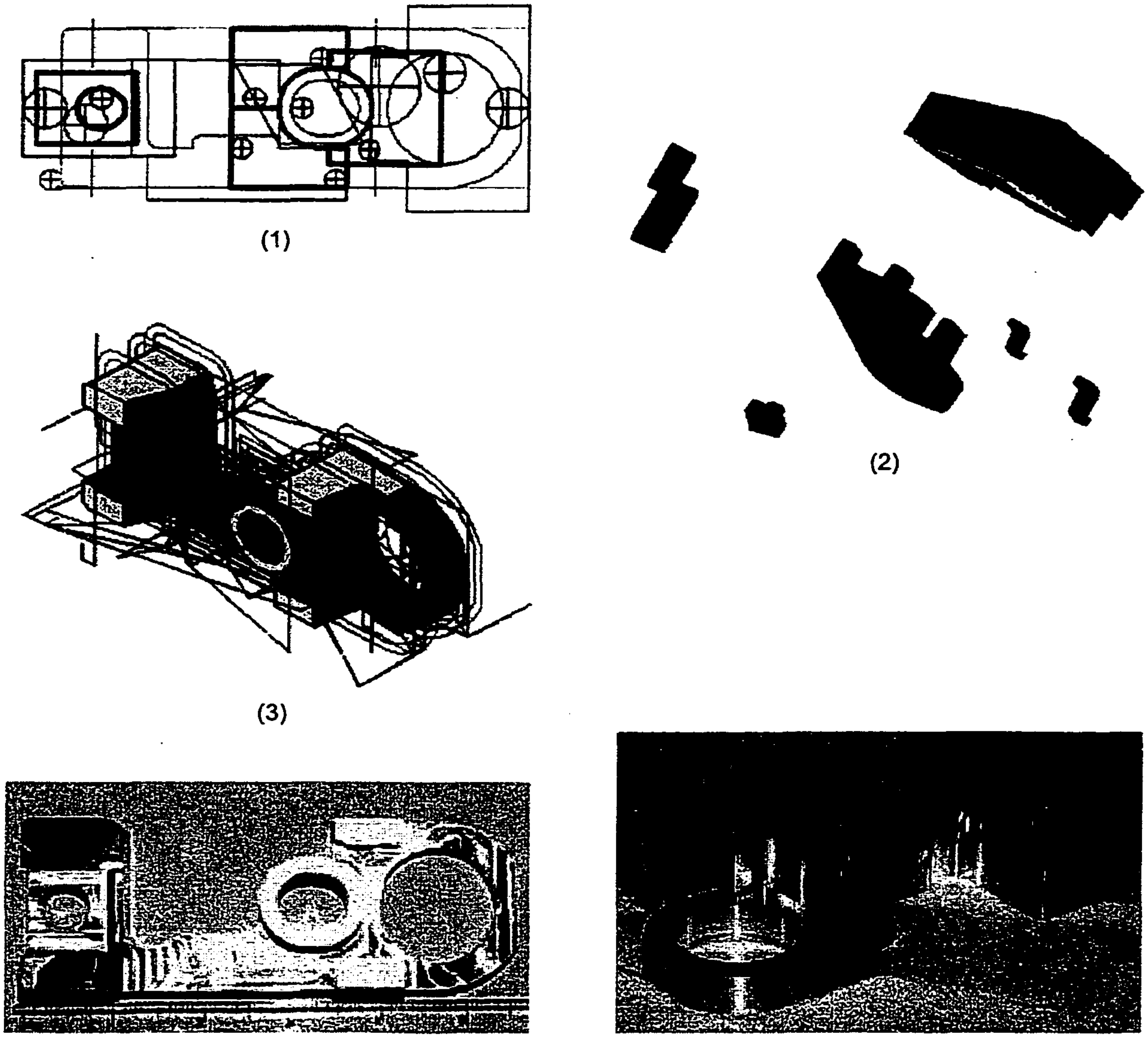
In Figures 7.1-7.4, different colors are used to indicate different NCML operations. The color scheme used is as follows:

- Yellow indicates the finished part or the original workpiece.

- Red, green and blue colors indicate NCML definitions or shapes caused from hole, contour, and pocket operations respectively.
- Violet and black is used for representing tool paths. Violet represents machining codes like G1, G2, and G3. Black indicates the rapid movement (G0).

Figures 7.1-7.4 show various prismatic shaped parts. The time required to generate detailed process plans was measured for some of the test parts, i.e. the time required to derive a complete set of machining instructions from a NCML file. These measured times will obviously vary with the skill of the operator but the relative values are still useful. Table 7.1 presents the time required for each step in the NCML fabrication process. The author assumed that there was an exact correspondence between the virtual tools defined in the NCML file and the physical tools available in the machining workcell, and furthermore that the order of operations specified in the file did not need to be altered. In actual practice, these assumptions may be incorrect and additional time would be required to choose appropriate substitute tools and/or change the order of operations.

One can see that the time required for process planning is quite short considering the complexity of the part shapes shown in Figures 7.1-7.4. The average time for the process planning was 18 minutes. This is considerably shorter than the conventional process. For instance, the Stone Machine Company typically plans on more than two hours for the process planning of a machined part with less complicated shapes. The information rich content of NCML reduced the burden on the operator in a number of ways. The Macro Simulation in FACILE allows for excellent visualization of both the finished part and the series of setups and operations required to achieve it. The automatic conversion of NCML into VRML provides a further visualization tool that makes Internet collaboration both possible and effortless. The hierarchical structure of NCML allows the operator to develop machining strategies which can be inherited from setups to patterns to operations and therefore contribute to timesavings in generating G-codes. An example part is presented in Appendix E with its associated NCML, VRML and G-code files.



(1)

(2)

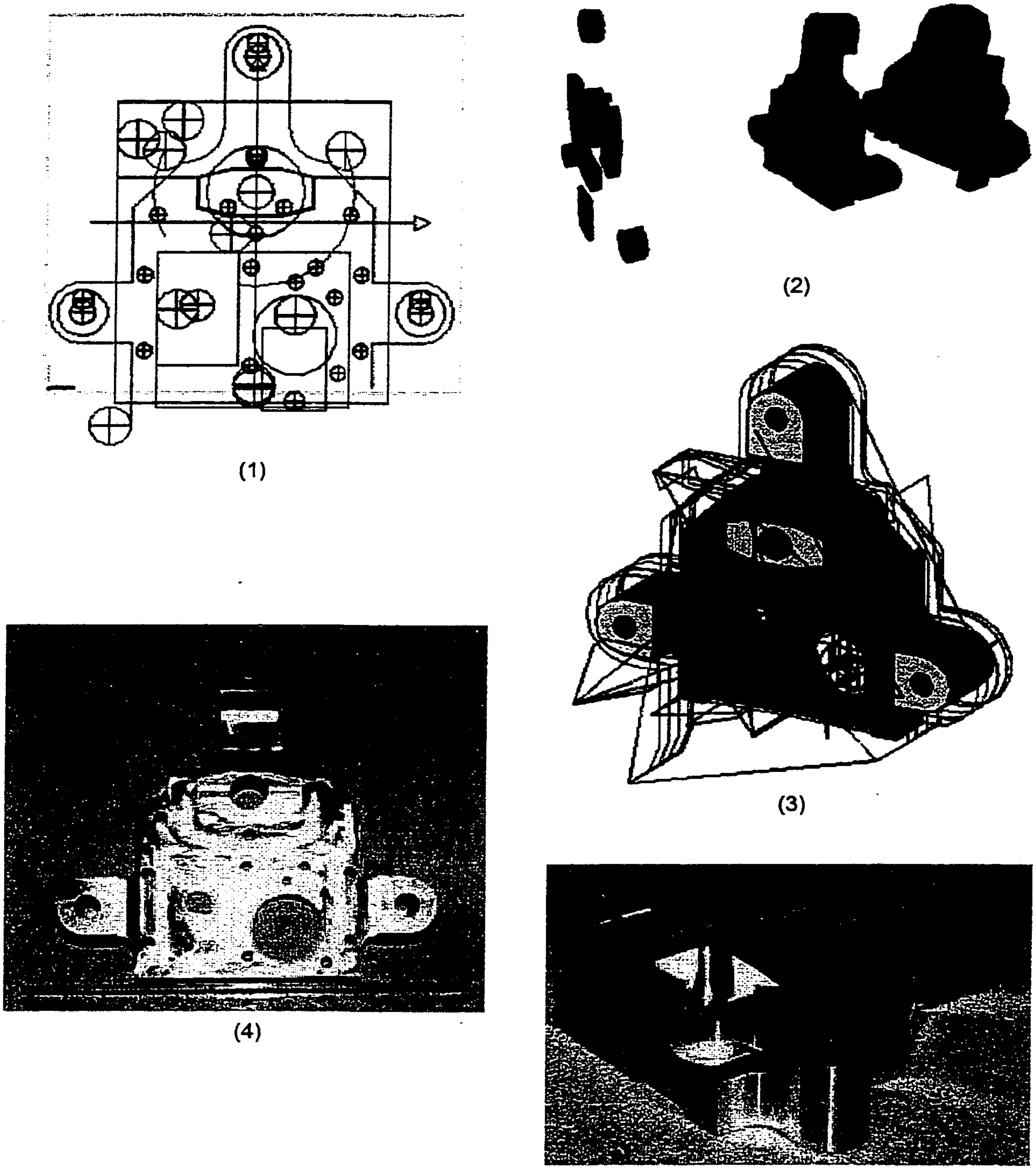
(3)

(4)

(5)

*Design source: National Design Repository
*Part name: CAD5[1].sat

Figure 7.1 Sample test part-1



(1)

(2)

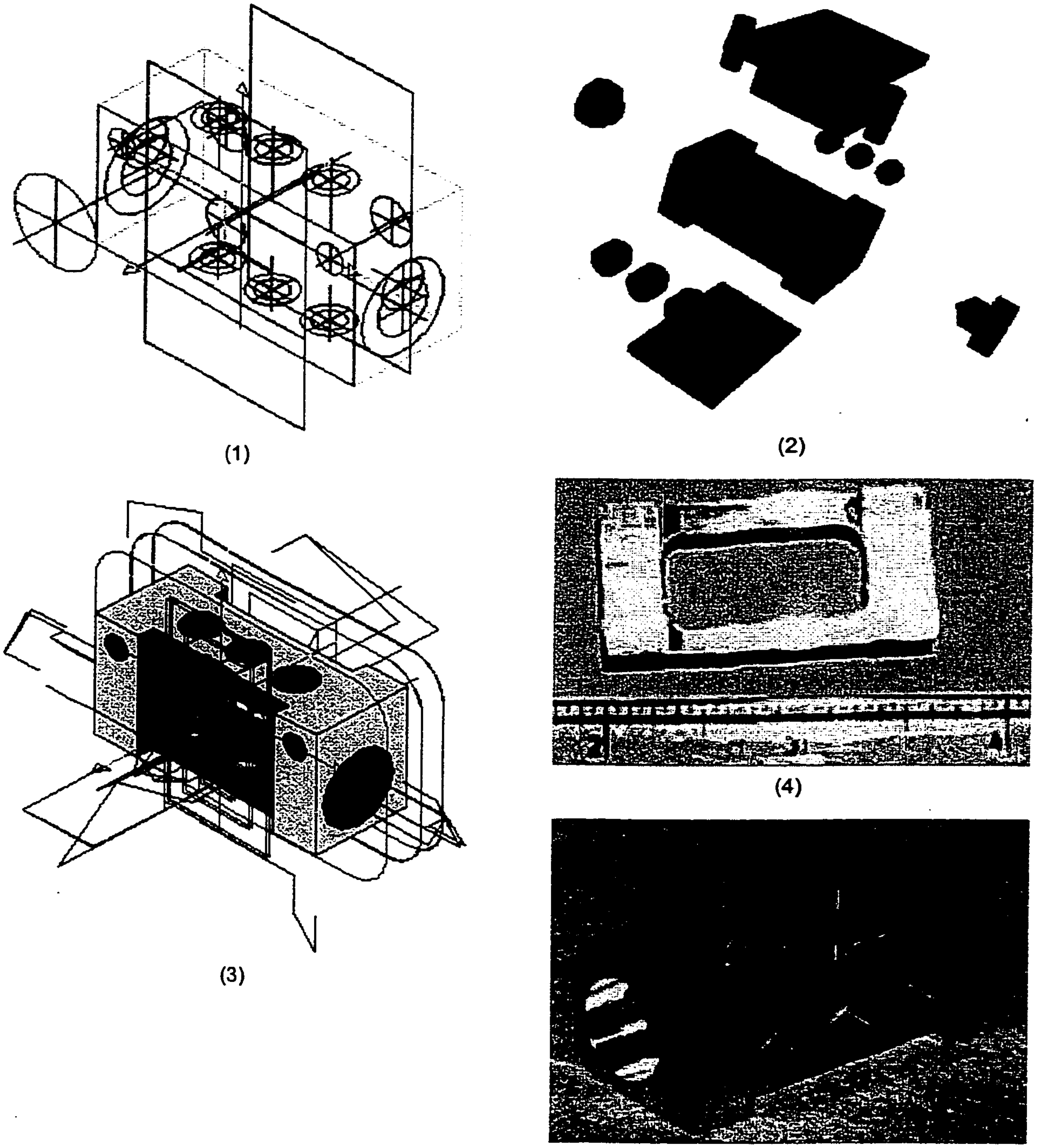
(3)

(4)

(5)

*Design source: National Design Repository
*Part name: 1797609.sat

Figure 7.2 Sample test part-2



(1)

(2)

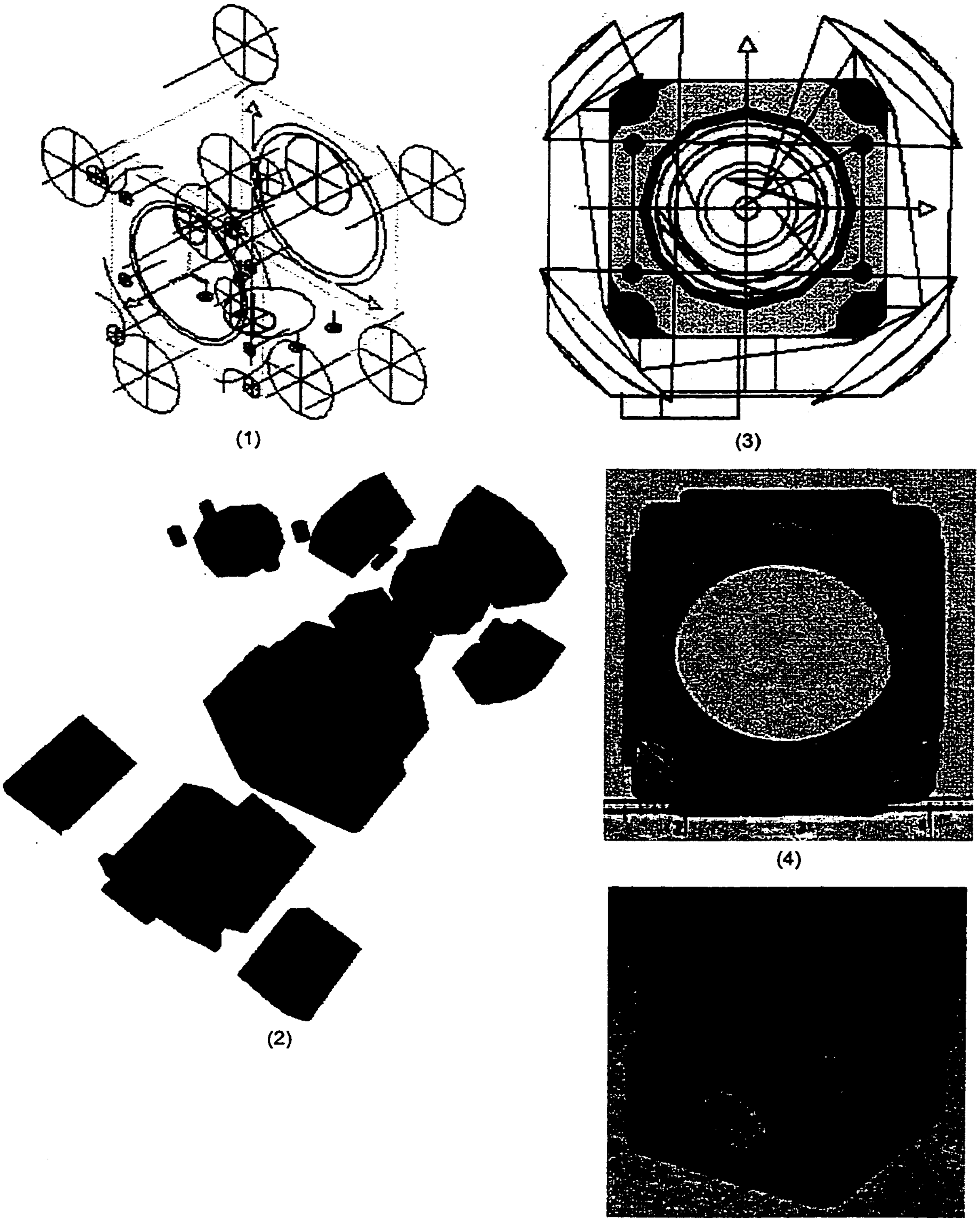
(3)

(4)

(5)

*Design source: MfgQuote.com
*Part name: RFQ-1093

Figure 7.3 Sample test part-3



(1)

(3)

(2)

(4)

(5)

*Design source: MfgQuote.com
*Part name: RFQ-1081

Figure 7.4 Sample test part-4

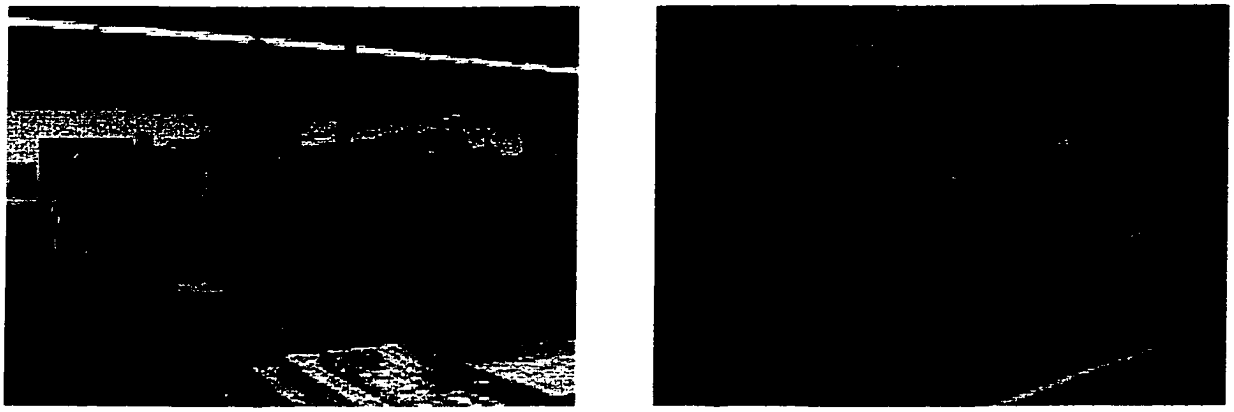
Table 7.1 Measured data from sample part fabrication

Source	Part name or File Name	Material	Overall size (cubic inches)	Number of setups	Number of patterns	Number of operations	NCML Creation (min)	Process Planning (min)	Machining (min)
Design Repository at Drexel Univ.	2638a.sat	Plastic	3.5*3.25*1	2	2	4	40	8	60
	CAD5[1].sat (See Figure 7.1)	Aluminum	5.5*2*2	4	10	17	170	20	210
	New_demo_us.sat	Plastic	5*3.25*2	2	8	21	60	13	100
	1797609.sat (See Figure 7.2)	Aluminum	5*5*1.2	2	10	39	120	20	300
MfgQuote.com	RFQ-1093 (See Figure 7.3)	Aluminum	1.7*0.8*0.6	6	14	24	130	15	200
	RFQ-1095	Aluminum	3.7*0.7*0.35	6	12	19	150	30	190
	RFQ-1081 (See Figure 7.4)	Aluminum	2.4*2.4*2	3	9	25	150	20	160
			Averages	3.6	9.3	21.3	117	18	174.3

Note:

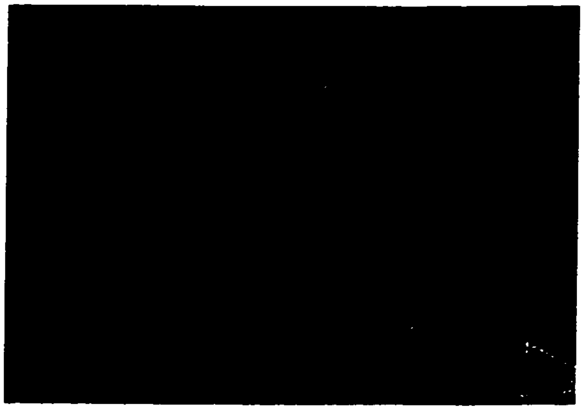
- The original file or part names used in the data source are kept for reference.
- The file extension "sat" indicates the file is an ACIS model.
- Machining time includes all activities of human and machine activities – setup, load/unload and cutting operations
- On the average, 22 minutes is spent for each machining setup. Therefore, cutting operation time can be calculated by, taking the total machining time and subtracting the number of setups multiplied by this average setup time.

Figure 7.5 A sample part used for quoting parameter calibration

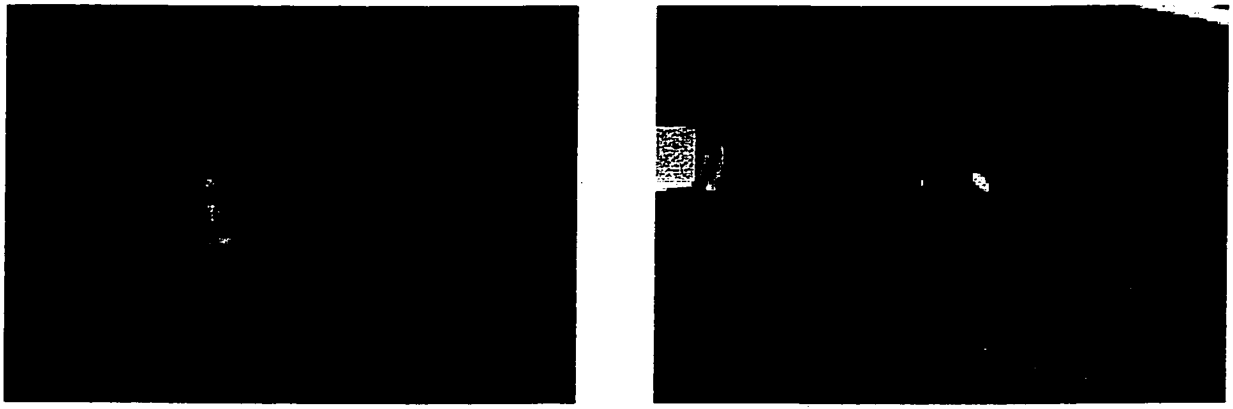


(a) HAAS F-4 Machining Center at the Stone Machine Company

(b) A machinst cleans coolant and metal chips on the fixture with pressed air.



(c) Work pieces are loaded on the machining center for fabrication. As seen in this photo, three parts are machined at a same time. Cutting operations follow after work piece loading



(d) A machinist unloads finished parts after cutting.

(e) A machinst conducts deburring with a manual device. Stacked parts are seen next to the machinist.

Figure 7.6 Machining observations at the Stone Machine Company

7.2 Quotation Helper Test

7.2.1 Parameter Calibration

Quotation parameter calibration was conducted at the Stone Machine Company. Four parts, seven machining setups, and more than sixty UMOs were observed. All parts were made from aluminum 6061-T6. Photos taken during the visits are presented here to illustrate the metal removal operations (see Figure 7.5 and Figure 7.6). As seen in Figure 7.6, machining a part is composed of two phases. The machinist-centered phases are where the machinist does most of the work, e.g. loading/unloading parts, clearing fixtures, and manual de-burring. The other phase is cutting, carried out by the machining center through execution of the G-codes. Normal human variability causes variations in the machinist centered phase and the associated parameters used in the Quotation Helper software may need to be increased to reflect this reality.

Parameters for calibrating the software are shown in Tables 7.2-3. The procedure for generating these parameters is more fully described in Appendix D.

Table 7.2 Basic calibrated quotation parameters for 6061 T6 Aluminum

Parameter	Values
MRR for Roughing	7.1 inch^3/Min
MRR for Finishing	66.4 inch^2/Min
MRR for Drilling	1.74 inch^3/Min
MRR for Facing	5.4 inch^2/Min
Tool Change Time	0.1 Min
Part Loading Time	1 Min
Part Quantity	1
Batch number	1
abor Cost	$60/Hour

Table 7.3 MRR adjustment factor for 6061 T6 Aluminum

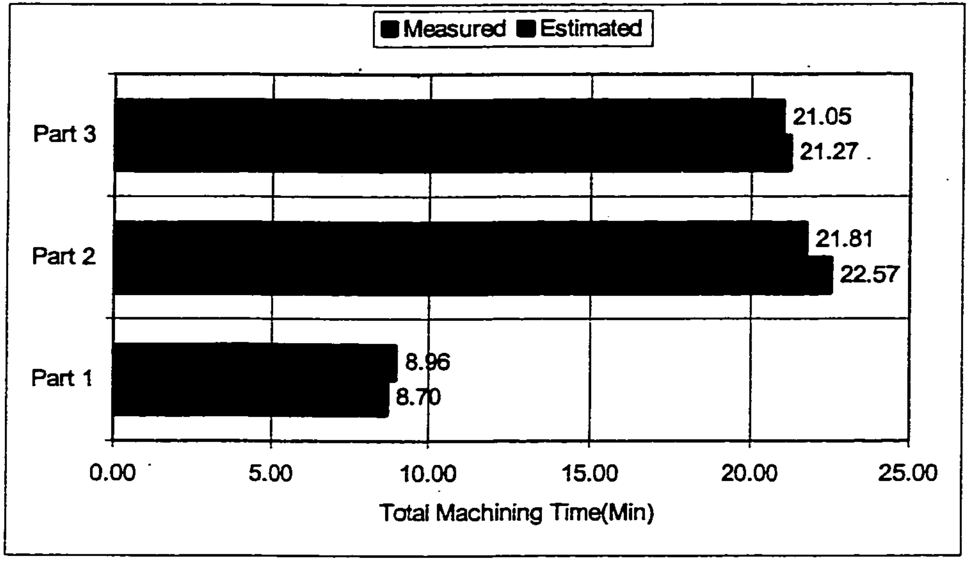
Sizing Tolerance (inch)	Weight factor*	Surface Finish Level	Weight factor*
~0.0005	0.01	~10	0.01
~0.002	0.03	~30	0.5
~0.005	0.37	~65	1
~0.01	1		

*The MRR is dependent on the variations in the size tolerance or the surface finish quality assigned to each NCML feature. This feature specific MRR is calculated by multiplying the corresponding weight factor by the original MRR. This relation is expressed in equation 5.7,5.8, and 5.9 in Chapter 5.

Table 7.4 Quotation helper test results

	Measured Time (Min)		Estimated Time (Min)*		Comparisons	
	Total Machining (T1)	Net Machining (N1)	Total Machining (T2)	Net Machining (N2)	T2/T1	N2/N1
Part 1	8.70	6.70	8.96	6.96	1.03	1.04
Part 2	22.57	18.07	21.81	14.31	0.97	0.79
Part 3	21.27	18.27	21.05	18.05	0.99	0.99

*Total Machining time includes Net Machining time and Machining preparation time (Loading/Unloading parts in fixtures)



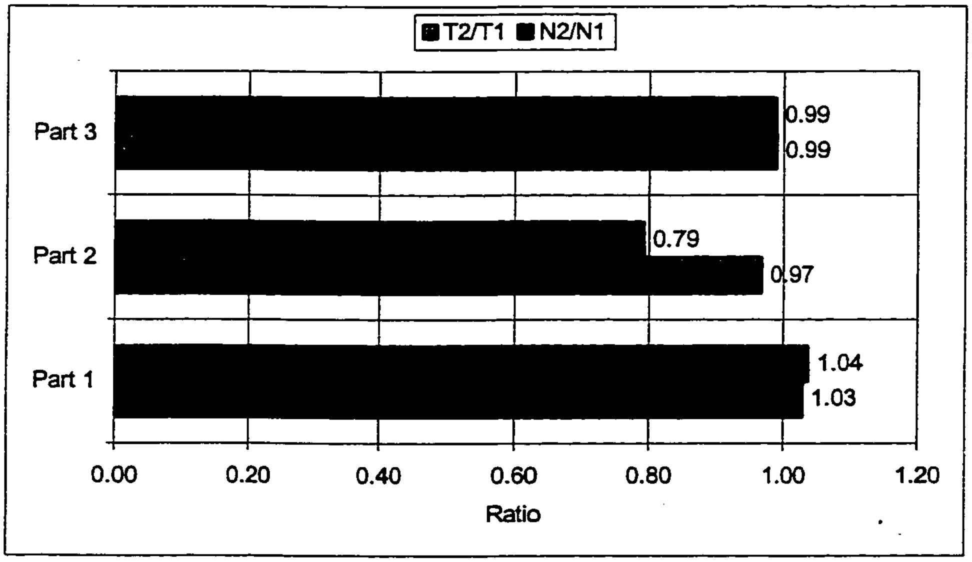


Figure 7.7 Comparison of estimation with measured machining time

7.2.2 Quotation Comparison Results

Estimated quotations are generated from the FACILE system for the three parts fabricated at the Stone Machine Company. The estimated machining time of each part is then compared to the actual machining time. In Figure 7.7, comparison results are given as a table and graphs. While the number of tested parts is admittedly small, the estimated times are quite close to the actual measured values. The estimated values average 99% of the total machining time, and 94% of the net machining time. Although more data is needed to prove the validity of the methodology, the experimental results illustrate the potential for accurate and fast cost estimation afforded by NCML.

7.2.3 Quote Range Test

We applied the quotation helper to parts posted on MfgQuote.com[1]. Since most of the parts are viewed and examined under a nondisclosure policy, only test results can be presented. Only the RFQ numbers used by MfgQuote.com are shown.

As seen in Chapter 5, the time required for quote estimation in FACILE from NCML is only a few minutes. This estimation process starts with a Macro Simulation which calculates volume and surface areas. The parameters in Tables 7.2-3 can then be used to quickly estimate machining times. The fixed costs associated with each RFQ are then added to the machining costs to obtain the estimate. Detailed procedures may be found in Appendix D.

Table 7.5 compares the quotation helper results with statistics obtained from the MfgQuote website. After each RFQ has been awarded, bidders are allowed to see the high, low, average and median quotes for the job which are compared to the estimates from FACILE. The wide range of bids is in itself quite interesting and illustrates the difficult nature of the business. Bidding too low on a job can be just as detrimental as bidding too high since most job shops don't like to lose money on their successful bids. The quotation helper enforces a discipline that should, at the very least, ensure some uniformity in the bidding process for the suppliers.

Five parts are tested and estimation values are as close as 81% of the average quotes and 97% of

---

[1] More information is available for MfgQuote.com. Readers are referred to Section 2.5, 6.2.1, and 6.2.2.

the median quote.

The results show that FACILE quotation helper can provide a reasonable valued quote for investigating parts fabrication requests.

Table 7.5 Quoting range test results

		Quoting Statistics ($).				NCML Estimation ($)		Comparison (%)	
Part ID	Qty.	Max	Min	Ave.(A)	Median(M)	Machining	Overall (E)	E/A	E/M
RFQ-884	250	6300	455	1736	1357	619	1319	75	97
RFQ-953	35	7599	654	2544	1795	1134	1458	57	81
RFQ-1081	25	2197	1050	1440	1351	653	959	67	71
RFQ-1113	2*10	1350	272	634	542	61	613	97	113
RFQ-1093	1	626	50	252	220	10	274	108	125
							Average	81	97

Note:

- Quoting statistics are from MfgQuote.com.
- Machining time includes all activities of human and machining operations –load/unload and cutting operations. The Overall (E) estimation includes fixture setup time in addition to machining time.
- Refer to Appendix D.2 for detailed quote calculation procedures used in getting values in the "Estimation" column.

7.3 NCML Modeling Capability Test

In NCML, the machined shape is expressed with simple geometric elements and various cutting tool shapes. These tests are intended to show how well the NCML representations can cover commonly used feasible shapes in the prismatic machined part domain. These tests compare design and manufacturing feature classifications.

Figure 7.8 illustrates machining feature shape classification. For additional explanation the reader is referred to the source publication (C-X Feng et al. [FKH96]).

Not all of these features can be expressed in the current implementation of NCML. Table 7.6 summarizes features which cannot be expressed or have some limitations in NCML.

With the use of the virtual tool concept, NCML can easily express fairly complicated machining features like the Dovetail slot, T-slot or Y-slot. However, any machining features that are built by rotating a section curve about any horizontal are not expressible in the current NCML definitions. This shows areas where NCML needs to be extended.

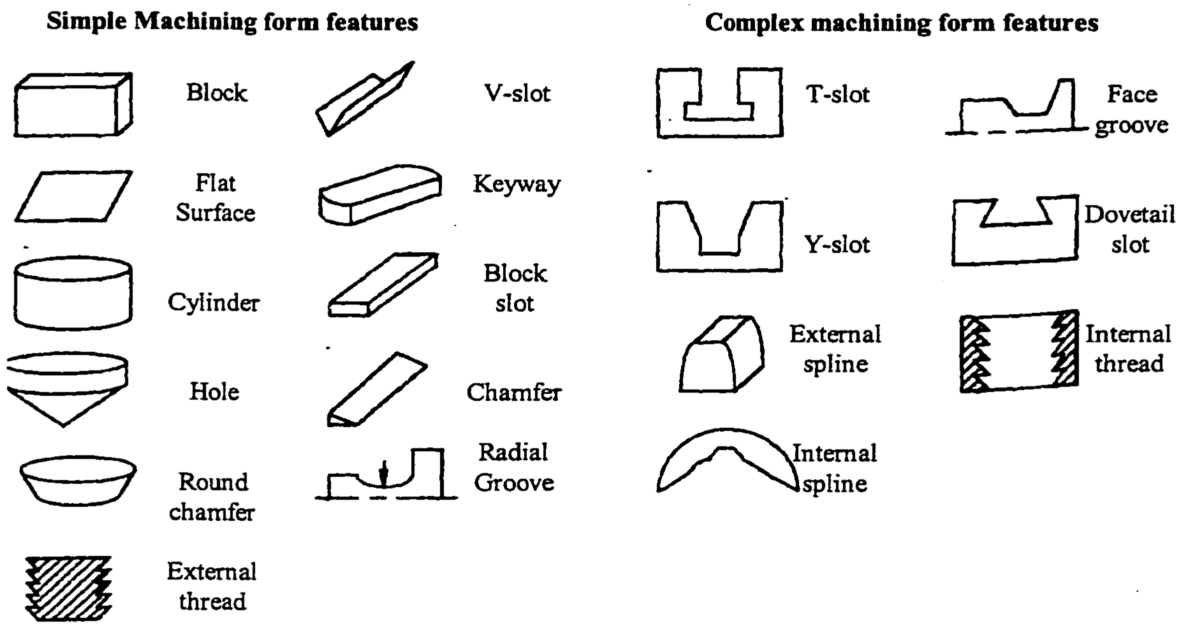
Typical Machining Form Features

Simple machining form features

- **block**
- **cylinder**
- **flat surface**
- **through hole**
- **blind hole**
- **round chamfer**
- **lengthy chamfer**
- **radial groove**
- **keyway**
- **block slot**
- **V- slot**
- **external thread**

Complex machining form features

- **face groove**
- **T - slot**
- **Y - slot**
- **dovetail slot**
- **internal spline**
- **external spline**
- **internal thread**



(Source: C-X Feng et al [FKH96])

Figure 7.8 Machining feature classification compared with NCML features

Table 7.6 Infeasible features in NCML

Feature Name	Comments
External thread	The rotational shape along any horizontal axis cannot be expressed in NCML. These features are likely to be machined by lathe operations not milling operations.
Radial groove	
Face groove	
External and Internal spline	If the spline geometry is composed of lines and arcs, NCML can express these form features. Currently, only line and arc geometry can be expressed in NCML. Therefore, NCML geometric representation needs to be added for more flexible curve geometry.

7.4 Process Improvements and Difficulties

There are a number of issues which require further consideration.

- Design aspects of NCML
  a. NCML creation currently relies on the design functions in FACILE/Design. Currently, there is no way to create NCML other than by using FACILE/Design. (Manual creation by using a text editor is possible but not practical). Automatic feature recognition to create NCML from a solid model would greatly enhance the usability of the system.

  b. The NCML based design representation is not used to describe the machined part itself, but to describe machined volumes that are supposed to be removed in machining operations. These volumes are determined by sweeping the machining tool along the feature geometry. It is therefore possible for the fabricator to assign different tools, which may lead to unintended results.

  c. Currently, NCML is equipped with only size tolerance information. This is inadequate to represent more complex GD & T information contained in the part design. Specifically, some tolerances that need to be defined based on other referential geometry cannot be handled.

- The quoting process using NCML is fast and showed fairly accurate results in the tests presented in this chapter. But the following issues should be remembered when NCML quoting.
  a. NCML quotation helper is more suitable for multiple quantity parts, not a single-quantity part. As seen in the RFQ-1093 case in Table 7.5, the estimated portion of quotation helper for a single quantity part is less than 10 % of the total estimate, because of the relatively large amount of time required for machine initialization.

  b. The quote is dependent on the process planning. There can be multiple ways of fabricating the same part. The order of setups and operations can affect the fabrication time. NCML uses the process plan given to the fabricator by the designer. Other alternative process plans made by machining experts may provide better results.

  c. Quoting based on the material removal rate may produce incorrect estimates. NCML quoting is dependent upon the machining volume, area and the cutting tool used. The NCML quotation may be inaccurate for the following reasons:

  i. Complexity of machining setup and operations. Quotation helper does not care about the relations between two machining volumes. For example, a sloped work piece setup and two intersecting holes need more attention in machining. The added attention means more fabrication time that can affect the quote.
  ii. Volume irrelevant operations like tapping holes. Some machining operation time may not be decided by its machining volume and area.
  iii. Detailed cutting tool planning provides better estimation. In the quotation helper, we assume that a single cutting tool is used for a specific machining operation. But in real situations, multiple tools are commonly used. For example, to cut a rectangular pocket with small radius corners, a large cutting tool may be used to cut out most of the material and a smaller tool is then used to finish the corners. This discrepancy may provide an overestimate of the cutting time since the small diameter cutter might be assumed to cut the whole pocket.

- Fabrication using FACILE/Fabricate: With the aid of a hierarchical data structure and inherited cutting conditions, process planning and tool path generation in FACILE/Fabricate can be quickly accomplished relative to the normal CAM software approach. However, the following performance issues may arise:
  a. G-code generation does not care about intersecting machining volumes. Therefore, unnecessary repetitive tool paths can be generated.
  b. Some of the NCML features cannot be handled in G-code generation. G-codes for a pocket with island geometry or non-convex pocket needs implementing in FACILE/Fabricator. This is purely an implementation issue, not a conceptual limitation of the system.

- Other needed extensions:
  a. Templates for machining strategies. FACILE/Fabricate processes NCML operations into necessary tool movements by assigning required machining strategy parameters in the process-planning phase (e.g. speeds, feeds, axial and radial depth of cut, number of roughing and semi-finishing passes, finish allowance, etc.). The machining strategy parameters corresponding to each NCML operation are given in Table C.2 in Appendix C. Currently, only one set of parameters for

each NCML operation is provided globally. By saving templates of machining strategies, the fabricator can reduce process-planning time by assigning exact conditions instead of manually entering each parameter value. A set of well-organized templates will really help machinists accumulate a base of experience for future use. Since the methods employed by fabricators vary tremendously, it is doubtful that these templates would be part of the NCML standard and are perhaps better left to the CAM or CNC Controller vendors to provide those templates when they process NCML operations.

b. Integrated Micro Simulation using tool paths (G-codes). It would be useful if FACILE/Fabricate were equipped with a built-in tool path simulation function. Users can simulate the resulting tool paths to check for errors. Based on the cutting simulation, machine time can be easily attained and this machining time can be compared with the machining time obtained from Quotation Helper. This will calibrate Quotation Helper parameters more efficiently and will provide more accurate quotations.

c. NCML representation of fixtures and detailed cutter properties for automatic feedrate optimization. See Section 8.1.2 for details

CHAPTER 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

8.1.1 General

This research project was initiated with the idea that the "Clean Interface" using the Mead-Conway method for VLSI chip manufacturing [Mead80] could be applied to mechanical parts. The specific application domain is fabricating prismatic parts that can be made with a standard NC machining center. The concept of machining/manufacturing features was adopted to represent the part geometry. This information is encoded using the current web document standard XML in a data format called NCML.

To prove the feasibility of NCML as a data interchange format, a number of test parts were created. Software systems were implemented to correspond with the core components of the proposed process. These computer software systems are FACILE/Design, E-Mill and FACILE/Fabricate. Machined part descriptions were created in the NCML format using FACILE/Design and hypothetically distributed to fabricators through E-Mill, a web-based e-commerce application. Several parts were actually fabricated with the aid of the process planning and G-code generation capabilities of FACILE/Fabricate.

The machining/manufacturing feature representations in NCML are the key elements to make a clean interface possible in the machined parts industry. The work accomplished for this dissertation is now summarized:

1. NCML, a new format based on XML, was devised as a machined part data interchange format between the designer and the fabricator. The fabricator can make parts in the specific domain without any information other than the NCML file. The contents of NCML are provided by the part designer in a format that meets the fabricator's requirements for conveniently manufacturing machined parts. The content of NCML reflects the fabricator's requirements and includes necessary and sufficient information for part fabrication. Besides geometric information, other components like machining features, tolerance, and simplified cutting tools were adopted in the NCML hierarchical structure to

satisfy the fabricator's requirements. NCML is an information-rich, unified data format compared to other geometric interchange formats. For example IGES files only handle the geometric aspect of the part, an essential but incomplete representation.

2. Based on NCML, a "clean interface" procedure for machined parts from design to manufacturing was devised. Compared to the conventional part manufacturing process, the suggested method provides both downstream (from the designer to the fabricator) and upstream (from the fabricator to the designer) information flows instead of only downstream information flows in the conventional process. In addition to the communication methods, the requirements of both the design and fabrication sides were determined. Macro simulation, Quotation Helper, Process planning, and G-code generation are the tasks typically required for the successful custom part fabrication using the NCML process.

3. E-Mill, a B2B e-commerce site, was developed to enable custom machined part fabrication business on the web. E-commerce environments provide a perfect test bed to verify the usability of NCML as a clean interface data format, since only a fully defined part specification could be used within e-commerce. E-Mill provides a VRML graphical representation of the NCML document and a markup capability for specific NCML elements, such that the buyer and the seller can exchange opinions. A competitive bidding process was also implemented in E-Mill. There are still many unsolved problems including user certification and the billing process. However, it illustrates that a NCML file could be used to deliver machined part information on the web where design and fabrication functions are physically separated.

4. When a large number of fabricators are bidding on the same part, the possibility of any one fabricator receiving the order is decreased. Therefore, an efficient method of generating quotes becomes necessary. With the aid of solid modeling capabilities, and the fabrication-centered, well-organized NCML structure, a simple quoting method was developed in this research project. Based on the material removal rate and the size of the cutting tool used, Quotation Helper in FACILE/Fabricate provides acceptable quote estimates. This feature can help fabricators estimate the part quote quickly and efficiently, thereby reducing the time spent in creating quotations.

5. A total of ten designs were obtained from three different sources and fabricated using the NCML process. During the fabrication process, the efficiency of process planning with NCML was evident. This efficiency is a product of the fact that the *operation* elements in NCML are equivalent to the actual machining operations like pocketing, contouring and hole making. The NCML hierarchical structure allows machining parameters to be inherited from *setups*, to *patterns*, and to *operation* elements.

6. All the activities and required functions mentioned in this dissertation were embodied into three computer software systems: FACILE/Design, E-Mill and FACILE/Fabricate. Even though these systems are not of commercial quality, they can provide interested researchers with sufficient functionality to validate the usability of NCML.

Critical commentary with regard to the components of the software systems is given in the following sections.

8.1.2 NCML

NCML is proposed as a basic enabling technology for the electronic buying and selling of custom machined parts. One of the most critical decisions made in this research was to choose XML for representing information. Therefore, NCML should be compatible with other web-based applications and establish a solid framework for future extension.

It is relatively easy to extend the definition of NCML. On the fabrication side, NCML could be used as an intermediate information model capturing all elements of the production process. An NCML file created by the designer can accumulate information at each fabrication activity and carry that information forward to the next task. Fixture descriptions, specialized tooling, material properties, and machine tool characterization are possible with future NCML extensions. Figure 8.1 shows a possible process with the extended NCML. This drawing assumes that NCML would be used with an automatic feed rate selection and on-line feed rate calibration. More specific material information and cutting tool parameters (flute information, cutter material, coating and parameters relating force to chip thickness) need to be employed in NCML for this purpose. Research in this area has been carried out at UNH [FJ98, FJH00, FJH99, FJHE99]. With this extension, NCML will serve as a main and integrated information source for the fabricator to proceed with operations at different stages of the machined part fabrication.

An effort to replace G-codes with a high-level language like NCML documents may also be worthwhile. G-codes are large in size but small in information content. They have no structural organization, and are difficult to edit. Correcting errors is laborious. Machine tool commands, G-codes, can be derived easily from a NCML file. If one can use OAC programmable and controllable capability, we might be able to drive a machine tool by sending a command directly from the NCML input. In such an environment, the NCML in Figure 8.1 will be able to play the role of providing information to the OAC application. This advanced future environment along with the extensions in Figure 8.1 would greatly improve the efficiency of machining.

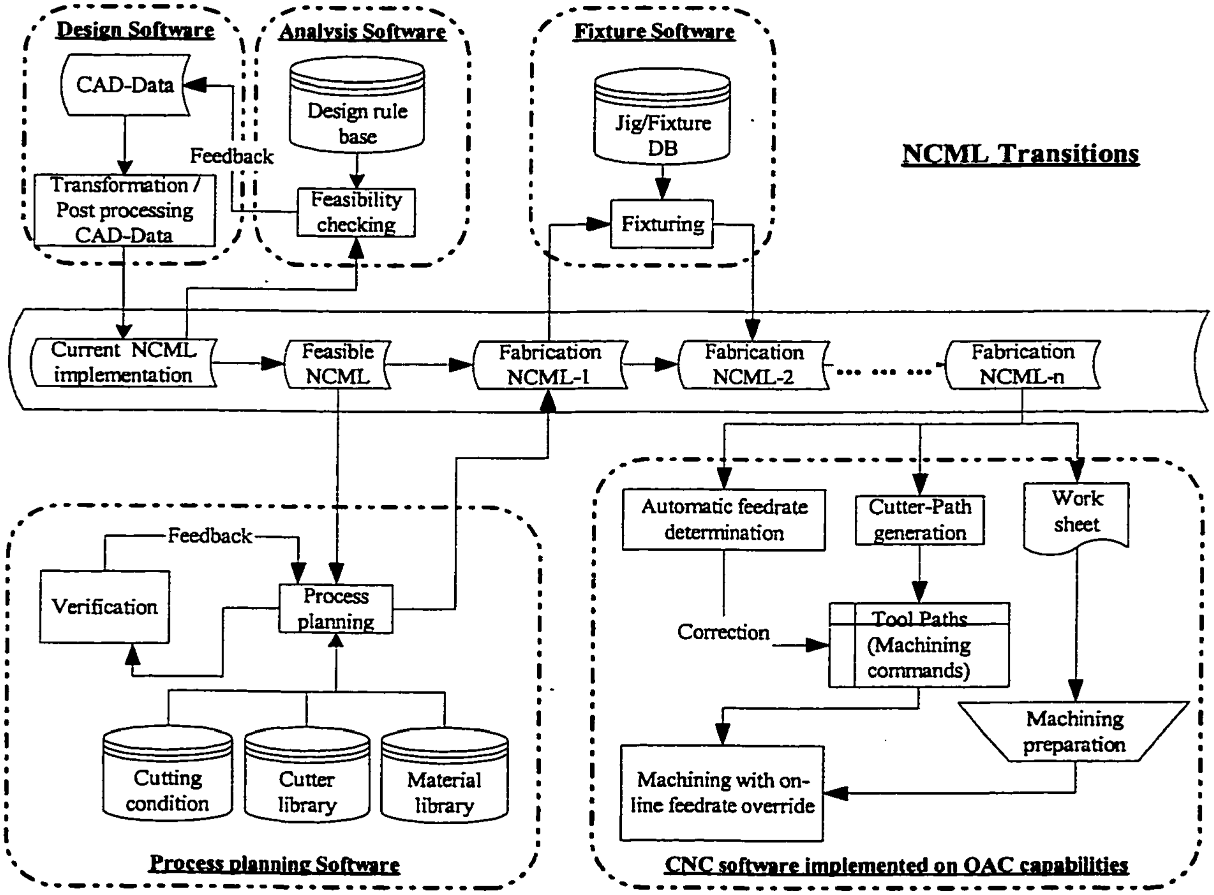


Figure 8.1 Application architecture with NCML extensions for the advanced machining environment

8.1.3 FACILE/Design

The software systems developed for this research are not robust enough to be commercially viable, and both the user interface and ease of use would need great improvement. But there are also some intrinsic problems that could not be solved by software developing. There are two issues related to the solid modeling kernel, Silver Engine from Schroff Development Corp [SDC, HOWE97]. FACILE/Design relies on solid modeling operations for its crucial functions like Macro Simulation. The solid modeling geometric kernel is Silver Engine, which sometimes generates operational errors due to unstable solid operations resulting in system crashes. The other potential problem is the frequent release of new API versions from Schorff Development Corp. Sometimes the API functions in the old version are not compatible with the new API specifications. It seems to be difficult to keep up with all the changes in APIs and frequent new versions.

The main purpose of FACILE/Design software is to create NCML documents. Therefore, the best way to attract actual designers to use NCML as their data interchange format, would be to make NCML files from their existing design software. This can be regarded as another form of data transformation or as a methodology for automatic feature recognition from the solid model. Creation of NCML directly from existing CAD software would greatly enhance the probability of success. Then FACILE/Design would only be used to add information to the extracted NCML or to edit operations.

8.1.4 E-Mill

The generation of a web-based application for e-commerce was very challenging, but there are a number of well established development tools and data standards that made the task somewhat easier. It is clear that the developer can be much more productive by using well established standards. The PHP language used for generating dynamic HTML files in a server side web application gave the author information on how web-based applications can be easily constructed.

The E-Mill implementation focused on the communication requirements for e-commerce in the metal removal industry. E-Mill is a prototype system, which lacks a number of essential elements. To make E-Mill a real e-commerce system capable of commercial application, it would be necessary to add a billing process, methods for user certification, and a part information confidentiality disclosure agreement.

Some functions in FACILE/Fabricate should be relocated in E-Mill. If E-Mill allows the fabricator to specify his or her machining capability as parameters, only screened feasible part information could be shown to the fabricator in E-Mill. Moreover, if Quotation Helper in FACILE/Fabricate is included in E-Mill, the fabricator will only see the feasible and targeted price-ranged parts automatically. Providing these fabrication side functions in the very early phase of commerce will save the fabricator time and enhance their competitive position.

8.1.5 FACILE/Fabricate

NCML provides just one plan for fabricating a part. This plan is not unique and it is possible to make any part with many different fabrication orders or different machining operations. For example, a circular shaped pocket may be fabricated as a large drilled hole. Depending on the machinist's experience they may have different preferences for machining operations and order of setups. There is therefore a need for functions that allow the user to change both the order and type of operations. Cost estimates of alternative ways of machining parts may be a part of picking alternative machining strategies.

The G-codes generated from FACILE/Fabricate are neither as efficient nor as robust as commercial CAM software systems. To make NCML a widely used part interchange standard, it would have to be linked to commercial CAM systems. Importing NCML into standard CAM software like MasterCam or FeatureCam may be a logical next step.

8.2 Recommendations for Future Research

The full potential of NCML as a design interchange format has yet to be reached. To make the concept become a reality there is a great deal of work to be done. The required tasks related to the current states of NCML are discussed below.

- Software Integration
  a. The translation of CAD geometry into NCML presents a challenge. It can certainly be done by

manual means, but automatic feature recognition and NCML generation directly from the CAD solid model would be preferable. Research by Nau [RGN94, RGN95], Shah [SMN94, SHMA95] may be applied to this task.

b. If NCML can be imported into commercial CAM systems, we can utilize the stable G-code generation capability of these commercial systems.

- Enrichment of Current functions
  a. The mapping of the NCML document into specific machining strategies and eventually into toolpaths should be fairly straightforward in most cases; however, the NCML document does not present a unique description of the part. Indeed, a particular part can be represented with an unlimited number of SETUPS, PATTERNS and OPERATIONS. The mapping from one NCML representation into others to accommodate the machinist's preferences and the capabilities of a particular type of NC machine is a worthwhile goal.
  b. There is no design checking function developed and the relevant designer or fabricator is charged with this task. Automatic or semi-automatic manufacturing feasibility checking will attract more users to use the NCML format. Geometric reasoning and fabrication rules should be applied together. See some previous works conducted by Wang and Wright [WW98].
  c. While creating G-codes from NCML in FACILE/Fabricate, it only processes the machining volume of the single operation and does not consider the relationship between operations, such as the intersection of one operation volume with another. G-code generation will need to consider feature interaction in order to prevent cutting the same area twice.

- Extension of NCML
  a. Everything that can be included on an engineering drawing, including dimensions, tolerances, finishing operations, notes and material characteristics must eventually be included in NCML. The tolerance-set element in NCML requires more development as it is not currently able to handle Geometric Form Tolerances or the tolerance relationships between machining features.
  b. Currently only a simple box shape is available as the initial work piece shape. It's common to use

a more versatile shape as the work piece starting point, e.g. a shape from casting. Referring to a solid model in a standard format (e.g. ACIS, Parasolids or STEP) might be a possible solution.

c. As shown in the previous chapter, some rotational machining form features cannot be expressed in NCML since NCML relies on the virtual tool concept and simple feature geometry. However, the rotational feature can be defined as a simple profile and additional parameters like the rotational angle (start and finishing angle) without any effect of the cutting tool. Adding new machining features is needed, while keeping consistency in the way of feature representations.

d. Features of sculptured surfaces do not belong to the focused domain in this dissertation. However, considering their common usage in industry, adding sculptured surface capability in NCML is a logical extension.

e. Representation of fixtures and specialized tooling is certainly desirable. Characterization of machine tools would aid in the matching of buyers and sellers since it will aid in the matching of the job with the capabilities of the Job Shop.

f. Extending NCML from the data interchange format between the designer and the fabricator to the working fabrication model is challenging. Since most engineering information on the fabrication side usually resides in different software formats, the information flow in the fabrication process is as difficult as the one between the designer and fabricator.

- Benchmarking

  a. The next phase of research is intended to test the NCML concept by applying it to a number of test parts obtained from the Drexel Standard Parts Repository.

  b. There are undoubtedly advantages and disadvantages of NCML compared to STEP Part AP224 and STEP-NC. A detailed comparison is necessary to determine if NCML is a viable competitor to these standards.

- Collaboration

Finally, the development of any standard requires the participation of many constituencies. The primary stakeholders in NCML include:

a. Job Shops

b. Machine Tool builders

c. Machine Tool Controller builders

d. CAM software vendors

e. Internet Broker service providers acting as a middleman for the matching of buyers and sellers of custom machined parts

f. CAD software vendors

To be successful NCML must meet the needs of all of these stakeholders.

APPENDIX A

NCML DTD

The following is the NCML DTD file which prescribes the contents and the structure of any NCML document. This file is composed of several "ELEMENT" and "ATTLIST" which regulate the contents and the structure of NCML.

```
<?xml version="1.0" encoding="UTF-8"?>
<!ENTITY % unit-type "(FEET|INCHES|METERS|CENTIMETERS|MILIMETERS)">
<!ENTITY % solid-type "(CUBE|CYLINDER|FILE)">
<!ENTITY % arc-direction-type "(CCW|CW)">
<!ENTITY % tool-type CUSTM|APT7P|DRILL|CTDRL|SPDRL|CBORE|CSINK|REAMT
|CREAM|KNURL|TAPPG|SIDEM|FLATM|BALLM|CNRRM|RNDEM|TPREM)">
<!ENTITY % tolerance-type "(GENERAL|GROUP)">
<!ELEMENT m-part (header, workpiece, structure?, body, tool-set,
tolerance-lib)>
<!ATTLIST m-part version CDATA #REQUIRED unit %unit-type; "INCHES">
<!ELEMENT header (part-spec, address?, memo?)>
<!ELEMENT workpiece (material, (cube | cylinder)?)>
<!ATTLIST workpiece type %solid-type; #REQUIRED source CDATA #IMPLIED>
<!ELEMENT structure (#PCDATA)>
<!ELEMENT body (setup+)>
<!ATTLIST body use-tolerance IDREF "TOL_-1">
<!ELEMENT tool-set (tool+)>
<!ELEMENT tolerance-lib (tolerance-set*)>
<!ELEMENT part-spec (name, date, revision)>
<!ELEMENT address ((name, tel, email), (company, street, region, postal-
code, locality, country)?)>
<!ELEMENT memo (#PCDATA)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT date (#PCDATA)>
<!ELEMENT revision (#PCDATA)>
<!ELEMENT tel (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT company (#PCDATA)>
<!ELEMENT street (#PCDATA)>
<!ELEMENT region (#PCDATA)>
<!ELEMENT postal-code (#PCDATA)>
<!ELEMENT locality (#PCDATA)>
<!ELEMENT country (#PCDATA)>
<!ELEMENT material (#PCDATA)>
<!ELEMENT cube (point, point)>
<!ELEMENT cylinder (point, point, diameter)>
<!ELEMENT point EMPTY>
<!ATTLIST point x CDATA "0.0" y CDATA "0.0" z CDATA "0.0" use CDATA
#IMPLIED>
<!ELEMENT diameter EMPTY>
```

```
<!ATTLIST diameter value CDATA #REQUIRED>
<!ELEMENT setup (axis, (pattern-hole | pattern-contour | pattern-
pocket)*)>
<!ATTLIST setup seq-no ID #REQUIRED use-tolerance IDREF "TOL_-1">
<!ELEMENT axis (point, point, point)>
<!ELEMENT pattern-hole (operation-hole+)>
<!ATTLIST pattern-hole seq-no ID #REQUIRED use-tool IDREF #REQUIRED use-
tolerance IDREF "TOL_-1">
<!ELEMENT pattern-contour (operation-contour+)>
<!ATTLIST pattern-contour seq-no ID #REQUIRED use-tool IDREF #REQUIRED
use-tolerance IDREF "TOL_-1">
<!ELEMENT pattern-pocket ((operation-pocket | operation-facing |
operation-boss)+)>
<!ATTLIST pattern-pocket seq-no ID #REQUIRED use-tool IDREF #REQUIRED
use-tolerance IDREF "TOL_-1">
<!ELEMENT operation-hole (depth, point)>
<!ATTLIST operation-hole seq-no ID #REQUIRED use-tolerance IDREF "TOL_-
1">
<!ELEMENT operation-contour (depth, offset, curve)>
<!ATTLIST operation-contour seq-no ID #REQUIRED use-tolerance IDREF
"TOL_-1">
<!ELEMENT operation-pocket (depth, curve+)>
<!ATTLIST operation-pocket seq-no ID #REQUIRED use-tolerance IDREF
"TOL_-1">
<!ELEMENT operation-boss (depth, curve+)>
<!ATTLIST operation-boss seq-no ID #REQUIRED use-tolerance IDREF "TOL_-
1">
<!ELEMENT operation-facing (depth, curve+)>
<!ATTLIST operation-facing seq-no ID #REQUIRED use-tolerance IDREF
"TOL_-1">
<!ELEMENT depth EMPTY>
<!ATTLIST depth value CDATA #REQUIRED>
<!ELEMENT offset EMPTY>
<!ATTLIST offset direction (NONE | RIGHT | LEFT) "NONE" value CDATA
"0.0">
<!ELEMENT curve ((point | line | arc), (line | arc)*)>
<!ELEMENT tool (diameter, flute-length?, corner-radius?, pitch-width?,
h-limit?, point-angle?, apt-parameter?, curve?, is-cutter?)>
<!ATTLIST tool index ID #REQUIRED type %tool-type; #REQUIRED>
<!ELEMENT flute-length EMPTY>
<!ATTLIST flute-length value CDATA #REQUIRED>
<!ELEMENT corner-radius EMPTY>
<!ATTLIST corner-radius value CDATA #REQUIRED>
<!ELEMENT pitch-width EMPTY>
<!ATTLIST pitch-width value CDATA #REQUIRED>
<!ELEMENT h-limit EMPTY>
<!ATTLIST h-limit value CDATA #REQUIRED>
<!ELEMENT point-angle EMPTY>
<!ATTLIST point-angle value CDATA #REQUIRED>
<!ELEMENT apt-parameter EMPTY>
<!ATTLIST apt-parameter  d-var CDATA "0.0" r-var CDATA "0.0" e-var CDATA
"0.0" f-var CDATA "0.0" a-var CDATA "0.0" b-var CDATA "0.0" h-var CDATA
"0.0">
<!ELEMENT is-cutter (bool+)>
<!ELEMENT bool EMPTY>
<!ATTLIST bool value (yes | no) "yes">
<!ELEMENT tolerance-set (bilateral?, surface-finish?, flatness?,
```

```
roundness?, profile?, concentricity?)>
<!ATTLIST tolerance-set index ID #REQUIRED type %tolerance-type;
"GENERAL" name CDATA "default_tol">
<!ELEMENT bilateral EMPTY>
<!ATTLIST bilateral value CDATA #REQUIRED>
<!ELEMENT surface-finish EMPTY>
<!ATTLIST surface-finish value CDATA #REQUIRED>
<!ELEMENT flatness EMPTY>
<!ATTLIST flatness value CDATA #REQUIRED>
<!ELEMENT roundness EMPTY>
<!ATTLIST roundness value CDATA #REQUIRED>
<!ELEMENT profile EMPTY>
<!ATTLIST profile value CDATA #REQUIRED>
<!ELEMENT concentricity EMPTY>
<!ATTLIST concentricity value CDATA #REQUIRED>
<!ELEMENT line ((point, point) | point)>
<!ELEMENT arc ((point, point, point) | (point, point))>
<!ATTLIST arc direction %arc-direction-type; "CCW" nvx CDATA "0.0" nvy
CDATA "0.0" nvz CDATA "0.0" >
```

APPENDIX B

NCML DTD MANUAL

This Appendix is devoted to the descriptions of each NCML element.

NCML General

- Version: 1.00
- Creator: Okhyun Ryou and Robert B. Jerard

What is the NCML DTD?

NCML is an XML data format designed for transferring part descriptions between designers and manufacturers through the Internet. Based on the destructive solid geometry (DSG), a NCML file describes how the manufacturing features would turn the raw block into the machined part in a standard 3-Axis machining center. To achieve a clean interface between the design and fabrication sides, the NCML DTD is developed to include sufficient fabrication information.
The NCML has the following merits compared to other design data interchange formats.

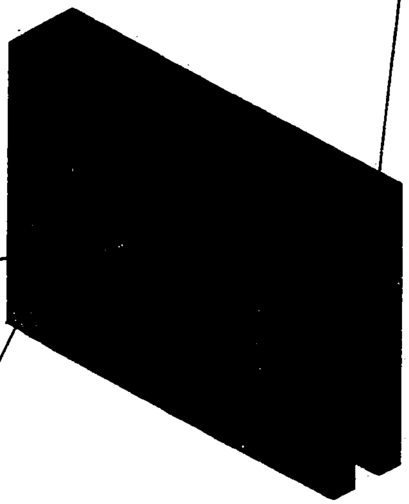
- NCML Structure - Machining Process planning - The machining features in the NCML are arranged into a hierarchical structure, which is intended to include sufficient information to plan the machining process.

- Manufacturing information - The NCML DTD is intended to include enough manufacturing data to minimize communication problems. Cutting tools and tolerance information are examples of transferring manufacturing related data. Moreover, most element names in the NCML are borrowed from the terms used in machine shops including setup, pattern, and operation.

A sample NCML document follows.

❑ A Sample Part in NCML

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE m-part SYSTEM
"d:\facile\lib\MPPML.dtd">
<m-part unit = "INCHES" version = "NCML(FACILE
Version 1.0)">
 <header>
  <part>
   <name>SAMPLE</name>
   <date>8/30/2000</date>
   <revision>Blank-Part Number</revision>
  </part>
  <address>
   <name>ROBERT B. JERARD</name>
   <tel>603)862-1234</tel>
   <email>ROBERT.JERARD@UNH.EDU</email>
  </address>
 </header>
 <workpiece type = "CUBE">
  <material>BLANK :Default Material</material>
  <cube>
   <point z = "-1.000000" use = "Minimum Point"/>
   <point x = "5.000000" y = "4.000000" use =
"Maximum Point"/>
  </cube>
 </workpiece>
 <body>
  <setup seq-no = "S:1">
   <axis>
    <point use = "ORIGIN"/>
    <point z = "1.000000" use = "Z-Axis"/>
    <point x = "1.000000" use = "X-Axis"/>
   </axis>
   <pattern-pocket seq-no = "S:1-PP:1" use-tool =
"T1">
    <operation-pocket seq-no = "S:1-PP:1-OP:1" >
     <depth value = "0.250000"/>
     <curve>
      <line>
       <point x = "4.000000" y = "3.000000"/>
       <point x = "1.000000" y = "3.000000"/>
      </line>
      <line>
       <point x = "1.000000" y = "1.000000"/></line>
      <line>
       <point x = "4.000000" y = "1.000000"/></line>
      <line>
       <point x = "4.000000" y = "3.000000"/></line>
     </curve>
    </operation-pocket></pattern-pocket>
   <pattern-hole seq-no = "S:1-PH:1" use-tool = "T2">
    <operation-hole seq-no = "S:1-PH:1-OP:1" >
     <depth value = "1.500000"/>
     <point x = "0.250000" y = "0.250000"/>
    </operation-hole>
    <operation-hole seq-no = "S:1-PH:1-OP:2" >
     <depth value = "1.500000"/>
     <point x = "4.750000" y = "0.250000"/>
    </operation-hole>
    <operation-hole seq-no = "S:1-PH:1-OP:3" >
     <depth value = "1.500000"/>
     <point x = "0.250000" y = "3.750000"/>
    </operation-hole>
    <operation-hole seq-no = "S:1-PH:1-OP:4" >
     <depth value = "1.500000"/>
     <point x = "4.750000" y = "3.750000"/>
    </operation-hole>
   </pattern-hole>
  </setup>
  <setup seq-no = "S:2">
   <axis>
    <point x = "5.000000" use = "ORIGIN"/>
    <point x = "1.000000" use = "Z-Axis"/>
    <point z = "-1.000000" use = "X-Axis"/>
   </axis>
   <pattern-contour seq-no = "S:2-PC:1" use-tool =
"T3">
    <operation-contour seq-no = "S:2-PC:1-OP:1" >
     <depth value = "0.500000"/>
     <offset direction = "NONE"/>
     <curve>
      <line>
       <point x = "0.500000" y = "-0.500000"/>
       <point x = "0.500000" y = "4.500000"/>
      </line></curve>
    </operation-contour></pattern-contour>
  </setup> </body>
 <tool-set>
  <tool type = "FLATM" index = "T1">
   <diameter value = "0.500000"/></tool>
  <tool type = "DRILL" index = "T2">
   <diameter value = "0.250000"/></tool>
  <tool type = "FLATM" index = "T3">
   <diameter value = "0.375000"/></tool>
 </tool-set>
 <tolerance-lib/></m-part>
```



❑ NCML DTD: Top Element Tree(s)

◆ m-part—The root element of any NCML document

```
m-part              //Root Element
|_(header,
| |_(part,
| | |_(name, date, revision )
| |__address?,
| | |_((name, tel, email),
| | |__(company, street, region, postal-code, locality, country)?)
| |__memo?)
|
|__workpiece,
| |_(material, cube | cylinder)?)
|
|__structure?,
|
|__body,
| |_(setup+)
|    |_(axis,
|    |__(pattern-hole |
|    |  |_(operation-hole+)
|    |  |   |_(depth, point) ...
|    |___pattern-contour |
|    |  |_(operation-contour+)
|    |  |   |_(depth, offset, curve) ...
|    |___pattern-pocket )* )
|       |_(operation-pocket| operation-facing| operation-boss)+
|          |_(depth, ... curve+) ...
|
|__tool-set,
| |_(tool+)
|    |_(diameter, flute-length?, corner-radius?, pitch-width?, h-limit?,
|      point-angle?, apt-parameter?, curve?,is-cutter?)
|                          |_((line | arc)*) ...
|
|__tolerance-lib )
|_(tolerance-set*)
|_(bilateral?, flatness?, roundness?, profile?, concentricity?)
```

Conventions:

"()": Parenthesis to include sub elements
"...": Element omission symbol
Element multiplicity symbols:
" ": Exactly One
"?": Optional (zero or one)
"+": One or more
"*": Many (zero or more)

Figure B.1 NCML element tree.

☐ NCML DTD Elements

- address—Part designer's address
- apt-parameter—APT seven parameter cutter geometry
- arc—An arc geometry
- axis—A local axis system(LCS)
- bilateral—Bilateral sizing tolerance range
- body—The structural element which includes ordered machining setups
- bool—Boolean values, yes or no
- company—The design company name
- concentricity—Concentricity tolerance range
- corner-radius—Cutting tool corner radius
- country—The country name
- cube—Cubic solid geometry
- curve—Curve geometry which consists of lines and arcs
- cylinder—Cylinder solid geometry
- date—Date
- depth—Machining depth value
- diameter—Cutting tool diameter
- email— The email address
- flatness—Flatness tolerance range
- flute-length—Cutting tool flute length
- h-limit—The h-limit of the tapping tool
- header—General information about the NCML part
- is-cutter—The cutting tool blade information
- line—Line geometry
- locality—Additional District name in the address element
- m-part—The root element of any NCML document
- material—The part material type
- memo—Comments
- name—The name of the designer or the part
- offset—Tool Offset value
- operation-contour—A contour following machining operation(feature)
- operation-boss—A boss cutting operation
- operation-hole—A hole making operation
- operation-facing—A facing operation
- operation-pocket—A pocket cutting operation
- part—Simple part description
- pattern-contour—The grouping element which includes operation-contour elements.
- pattern-hole—The grouping element which includes operation-hole elements
- pattern-pocket—The grouping element which includes operation-pocket elements
- pitch-width—The pitch length of the tapping tool
- point—Point definition
- point-angle—The tool tip angle of the cutting tool
- postal-code—The postal code (ZIP code in the U.S.A.)
- profile—Profile tolerance range
- region—State name or major district name inside an address element
- revision—Part number and part revision information
- roundness—Roundness tolerance range
- setup—Machining setup information.
- street—Street, or town name inside the address element

- structure—An optional element displaying the structure of machining operations.
- tel—Phone number
- tolerance-lib—An element which groups tolerance-set elements
- tolerance-set—A set of actual tolerance ranges
- tool—A cutting tool definition
- tool-set—An element which groups tool elements
- workpiece—geometric and specific information about the raw work piece

☐ NCML Element Descriptions

◆ address

Address information. Fabricators can contact the machine part designer with this element.

Content
CONTENT DECLARATION: ((name,tel,email), (company,street,region,postal-code,locality,country)?)

- company—The design company name
- country—The country name
- email— The email address
- locality—Additional district name in the address element
- name—The name of the designer or the part
- postal-code—The ZIP code
- region—State name or major district name inside an address element
- street—Street, town name inside the address element
- tel—Phone number

Attributes
None

Parent Elements

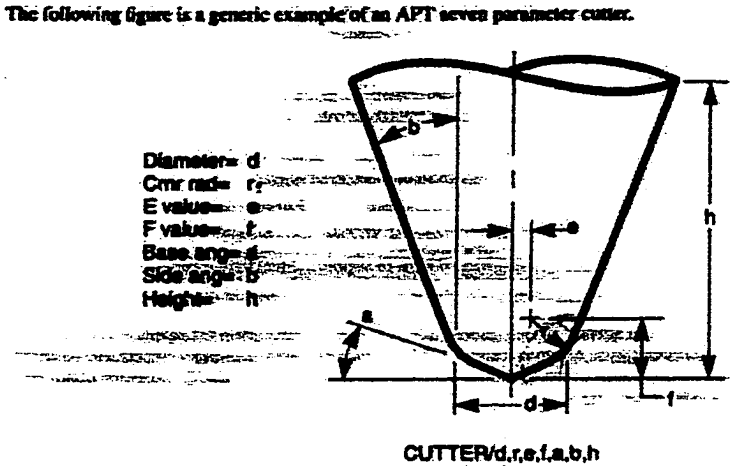
- header—General information about the NCML part

◆ apt-parameter

APT seven-parameter cutter geometry.

Content
Empty

Attributes
✓ *a-var*
Data type: double, Value(s): CDATA, Default Value: 0.0
✓ *b-var*
Data type: double, Value(s): CDATA, Default Value: 0.0
✓ *d-var*
Data type: double, Value(s): CDATA, Default Value: 0.0
✓ *e-var*
Data type: double, Value(s): CDATA, Default Value: 0.0
✓ *f-var*
Data type: double, Value(s): CDATA, Default Value: 0.0
✓ *h-var*
Data type: double, Value(s): CDATA, Default Value: 0.0
✓ *r-var*
Data type: double, Value(s): CDATA, Default Value: 0.0



(Source: VERICUT 2.0 User Manual [CGT94])

Figure B.2 The APT seven-parameter cutter representation.

Parent Elements

- tool—A cutting tool definition

♦ arc

Arc definition. Three points and additional attributes determine an arc. Three points are starting, arc center and end-points of an arc. The first point can be omitted when the first point is shared with the end point of the previous element.

Content

CONTENT DECLARATION: ((point,point,point)| (point,point))

- point—Point definition

Attributes

Three attributes - nvx, nvy and nvz - define a vector perpendicular to the arc definition plane. The direction attributes determines the arc orientation according to the vector.

✓ *direction*
Arc Direction has two options: CCW(Counter-Clockwise) CW(Clockwise)
Vector x coordinate value. Type: double
Vector y coordinate value. Type: double
Vector z coordinate value. Type: double
Value(s): CCW/CW, Default Value: CCW
✓ *nvx*
Value(s): CDATA, Default Value: 0.0
✓ *nvy*
Value(s): CDATA, Default Value: 0.0
✓ *nvz*
Value(s): CDATA, Default Value: 0.0

Parent Elements

- curve—Curve geometry which consists of lines and arcs

◆ axis

Defines an Local Coordinate System (LCS). Three vector (point) elements are enough to define an LCS in terms of the Global Coordinate System (GCS). Three vectors are required: the origin of the LCS, an x direction vector and a z direction vector in that order. (y direction is implicit since it is an orthogonal right hand coordinate system.

Content
CONTENT DECLARATION: (point,point,point)

- point—Point definition

Attributes
None

Parent Elements

- setup—Machining setup information.

◆ bilateral

Sizing tolerance. This is a general tolerance specification for the size and position of features.

Content
Empty

Attributes
✓ *value*
This field value ranges from negative to positive input values. Therefore, the value 0.005 means that the bilateral tolerance is between -0.005 and +0.005.
Type: double, Value(s): CDATA, Default Value: #REQUIRED

Parent Elements

- tolerance-set—A set of tolerance ranges

◆ body

The body element contains information about fabrication processes and machining features. This element is one of the top-level elements in an NCML document and is composed of at least one SETUP element. The BODY element (structure) must appear in the top level of the file only once.

Content
CONTENT DECLARATION: (setup+)

- setup—Machining setup information.

Attributes
✓ *use-tolerance*
Value(s): IDREF, Default Value: TOL_-1

Parent Elements
m-part—The root element of any NCML document

◆ bool

Boolean variable

Content
Empty

Attributes
✓ *value*
There are two possible values: yes (true) and no (false)
Value(s): yes or no, Default Value: yes

Parent Elements

- is-cutter—The cutting tool blade information

◆ company

Company or any organization name

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements

- address—Part designer's address

◆ concentricity

Concentricity tolerance. This element indicates the tolerance zone of the axial deviation of two concentric holes or surfaces of revolution.

Content
Empty

Attributes
✓ *value*
This field value ranges from negative to positive input values.
Type: double, Value(s): CDATA, Default Value: #REQUIRED

Parent Elements

- tolerance-set—A set of actual tolerance ranges

◆ corner-radius

The corner radius of a cutting tool.

Content
Empty

Attributes
✓ *value*
Data type: double, Value(s): CDATA, Default Value: #REQUIRED

Parent Elements

- tool—A cutting tool definition

◆ country

Country Name

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements

- address—Part designer's address

◆ cube

Cube solid geometry. This element is composed of two point elements. The first point indicates the coordinates of the corner with minimum values of a box and the second one indicates the coordinates of the corner with maximum values of a box.

Content
CONTENT DECLARATION: (point,point)

- point—Point definition

Attributes
None

Parent Elements

- workpiece—Raw material block information

◆ cylinder

Cylinder geometry. This element is composed of two point elements that determine the centerline of a cylinder and a value for the diameter.

Content
CONTENT DECLARATION: (point,point,diameter)

- diameter—Cutting tool diameter
- point—Point definition

Attributes
None

Parent Elements

- workpiece—Raw workpiece information

◆ date

Date Information

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements

- part—Simple part description

◆ depth

Cutting depth.

Content
Empty

Attributes
✓ *value*
Data type: double, Value(s): CDATA, Default Value: #REQUIRED

Parent Elements

- operation-boss—A boss cutting operation
- operation-contour—A contour following machining operation (feature)
- operation-facing—A face cutting operation
- operation-hole—A hole cutting operation
- operation-pocket—A pocket cutting operation

◆ diameter

Diameter.

Content
Empty

Attributes
✓ *value*
Data type: double, Value(s): CDATA, Default Value: #REQUIRED

Parent Elements
- cylinder—Cylinder solid geometry
- tool—A cutting tool definition

◆ email

Email address

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements
- address—Part designer's address

◆ flatness

Tolerance for surface flatness.

Content
Empty

Attributes
✓ *value*
This field value range from negative to positive input values. Type: double
Value(s): CDATA, Default Value: #REQUIRED

Parent Elements
- tolerance-set—A set of actual tolerance ranges

◆ flute-length

The flute length of the tool. This field is only valid if the attribute type is set to the SIDEM (Side End Mill) in the tool element.

Content
Empty

Attributes
✓ *value*
Data type: double, Value(s): CDATA, Default Value: #REQUIRED

Parent Elements
- tool—A cutting tool definition

◆ header

General information on the NCML document. The header element contains the general information of the part specification and the part designer.

Content
CONTENT DECLARATION: (part,address?,memo?)
- address—Part designer's address
- memo—Comments
- part—Simple part description

Attributes
None

Parent Elements
m-part—The root element of any NCML documents

◆ h-limit

The H LIMIT value which is one of the tapping tool properties. This field is valid only in case that the tool type is TAPPG (Tapping Tool)

Content
Empty

Attributes
✓ *value*
Data type: integer
Value(s): CDATA, Default Value: #REQUIRED

Parent Elements

- tool—A cutting tool definition

◆ is-cutter

Cutting edge expression. The array of Boolean elements indicates the cutting edges of the tool section. The number of Boolean values depends on the number of segments in the curve element. "yes" values of the bool elements mean the corresponding segment has a cutting edge. This is valid only if the tool type attribute is set to "CUSTM" type.

Content
CONTENT DECLARATION: (bool+)

- bool—Boolean values, yes or no

Attributes
None

Parent Elements

- tool—A cutting tool definition

◆ line

Line definition. Two points determine a line element. The first point can be omitted when the line is used in curve element and the first point is shared with the end point of the previous element.

Content
CONTENT DECLARATION: ((point,point)|point)

- point—Point definition

Attributes
None

Parent Elements

- curve—Curve geometry which consists of lines and arcs

◆ locality

Additional District name

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements
- address—Part designer's address

◆ m-part

The root element of any NCML document. This is used to include all NCML elements inside one element.

Content
CONTENT DECLARATION: (header,workpiecee,structure?,body,tool-set,tolerance-lib)
- body—The structural element which includes ordered machining setups
- header—General information about the NCML part
- structure—An optional element displaying the structure of machining operations.
- tolerance-lib—An element which groups "tolerance-sets"
- tool-set—An element which groups tool elements
- workpiece—Raw workpiece information

Attributes
✓ *unit*
The length unit used in the NCML fle
Value(s): FEET/INCHES/METERS/CENTIMETERS/MILIMETERS, Default Value: INCHES
✓ *version*
The NCML Version
Value(s): CDATA, Default Value: #REQUIRED

Parent Elements
None

◆ material

Material Name. Type - Simple data element.

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements
- workpiece—Raw workpiece information

◆ memo

Comments on the part described in the NCML document.

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements
- header—General information about the NCML part

◆ name

Identification of a designer or a part.

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements
- address—Part designer's address
- part—Simple part description

◆ offset

Tool offset definition

Content
Empty

Attributes
✓ *direction*
Cutting tool offset direction. This attribute can be set to one of three predefined values below. In case of NONE, the tool center moves along the tool path. In case of LEFT (RIGHT), the tool moves left (right) and perpendicular to the tool paths by half of its diameter in order to follow the path by its side.
Value(s): NONE/RIGHT/LEFT, Default Value: NONE
✓ *value*
Offset quantity in the defined length unit. Data type: double
Value(s): CDATA, Default Value: 0.0

Parent Elements
- operation-contour—A contour following machining operation(feature)

◆ operation-boss

Boss cutting operation. The element describes the geometry of a boss operation. The depth element addresses the pocket machining depth. This element can include more than one closed curve element for boss geometry. The cutting tool information and machining setup information are inherited from the pattern-pocket element that includes this operation-pocket element. The tolerance-set information can be inherited from the encompassed elements or can be specified with a use-tolerance attribute

Content
CONTENT DECLARATION: (depth, curve)
- depth—Machining depth value
- curve—Curve geometry which consists of lines and arcs

Attributes
✓ *seq-no*
Value(s): ID, Default Value: #REQUIRED,
✓ *use-tolerance*
Value(s): IDREF, Default Value: TOL_-1

Parent Elements
- pattern-pocket—The grouping element which includes operation-pocket elements

◆ operation-contour

Contour-machining operation. The operation-contour element describes the path of a machining operation. In addition to contour geometry, the tool offset and depth give more flexibility to define various contour-following operations. The cutting tool information and machining setup information are inherited from the pattern-contour element that includes this operation-contour element. The tolerance-set information can be inherited from the encompassed elements or can be specified with the use-tolerance attribute. The operation-contour element must be placed inside the pattern-contour element.

Content
CONTENT DECLARATION: (depth, offset, curve)

- depth—Machining depth value
- offset—Tool offset value
- curve—Curve geometry which consists of lines and arcs

Attributes

✓ *seq-no*
Value(s): ID, Default Value: #REQUIRED,
✓ *use-tolerance*
Value(s): IDREF, Default Value: TOL_-1

Parent Elements

- pattern-contour—The grouping element which includes operation-contour elements.

♦ operation-facing

Face-cutting operation. This element describes the geometry of a plane. The depth element addresses the pocket machining depth. This element can include one-curve element that determines the plane geometry in 3D space. Cutting tool information and machining setup information are inherited from the pattern-pocket

Content

CONTENT DECLARATION: (depth, curve)

- depth—Machining depth value
- curve—Curve geometry which consists of lines and arcs

Attributes

✓ *seq-no*
Value(s): ID, Default Value: #REQUIRED,
✓ *use-tolerance*
Value(s): IDREF, Default Value: TOL_-1

Parent Elements

- pattern-pocket—The grouping element which includes operation-pocket elements

♦ operation-hole

Hole-making operation. The operation-hole element specifies hole location and depth. Cutting tool information and machining setup information are inherited from the pattern-hole element, which includes the operation-hole element.

Content

CONTENT DECLARATION: (depth, point)

- depth—Machining depth value
- point—Point definition

Attributes

✓ *seq-no*
Value(s): ID, Default Value: #REQUIRED,
✓ *use-tolerance*
Value(s): IDREF, Default Value: TOL_-1

Parent Elements

- pattern-hole—The grouping element which includes operation-hole elements

♦ operation-pocket

Pocketing operation. This element describes the geometry of a pocket with several islands. The depth element addresses the pocket machining depth. This element can include at least one curve element. The first curve describes the pocket boundary geometry and the others describe the island boundary geometry. The cutting tool information and machining setup information are inherited from the pattern-pocket element that includes this operation-pocket element.

Content
CONTENT DECLARATION: (depth, curve+)

- depth—Machining depth value
- curve—Curve geometry which consists of lines and arcs

Attributes
✓ *seq-no*
Value(s): ID, Default Value: #REQUIRED,
✓ *use-tolerance*
Value(s): IDREF, Default Value: TOL_-1

Parent Elements

- pattern-pocket—The grouping element which includes operation-pocket elements

◆ part

Simple Machined part description.

Content
CONTENT DECLARATION: (name, date, revision)

- date—Date
- name—The name of a designer or a part
- revision—Part number and Part Revision information

Attributes
None

Parent Elements

- header—General information about the NCML part

◆ pattern-contour, pattern-hole, pattern-pocket

Grouping operation elements. These elements can include more than one of the same types of operation elements. In NCML, machining operations should be grouped by machining type (pattern-* element) and the tool that may be used in the real machining operation. The pattern-* elements are designed for this grouping purpose. A tool index number and at least one operation element should be included in the pattern-* element according to the type of the pattern elements. The tolerance-set information is maybe inherited from the setup element or defined in this element with USE_TOLERANCE attribute.

Content
CONTENT DECLARATION: (operation-contour+)

- operation-contour—A contour following machining operation(feature)

Attributes
✓ *seq-no*
Value(s): ID, Default Value: #REQUIRED
✓ *use-tolerance*
Value(s): IDREF, Default Value: TOL_-1
✓ *use-tool*
The ID reference of the tool element. The value of the tool to be used should exist in the tool element.
Value(s): IDREF, Default Value: #REQUIRED

Parent Elements

- setup—Machining setup information.

◆ pitch-width

Pitch length. This field is valid only in case that the tool type is TAPPG (Tapping Tool)

Content
Empty

Attributes
✓ *value*
Data type: double, Value(s): CDATA, Default Value: #REQUIRED

Parent Elements

- tool—A cutting tool definition

◆ point

Point definition

Content
Empty

Attributes
x coordinate value. Type: double
y coordinate value. Type: double
z coordinate value. Type: double
✓ *use*
Optional attribute.
This attribute can be used to add explanations for point usage.
Value(s): CDATA Default Value: #IMPLIED
✓ *x*
Value(s): CDATA, Default Value: 0.0,
✓ *y*
Value(s): CDATA, Default Value: 0.0
✓ *z*
Value(s): CDATA, Default Value: 0.0

Parent Elements

- arc—Arc geometry
- axis—The local axis system(LCS)
- cube—Cubic solid geometry
- cylinder—Cylinder solid geometry
- line—Line geometry
- operation-hole—A hole cutting operation
- curve—Curve geometry which consists of lines and arcs

◆ point-angle

Cutting tool tip angle. The angle of the cutter tip measured from the horizontal plane. This field is valid only in the case that the TYPE is set to one of CTDRL (Center Drill), SPDRL (Spot Drill), CSINK (Counter Sink), CREAM (Center Reaming Tool), TAPPG (Tapping Tool), or TPREM (Tapered end mill).

Content
Empty

Attributes
✓ *value*
Data type: degree
Value(s): CDATA, Default Value: #REQUIRED

Parent Elements

- tool—A cutting tool definition

◆ curve

A curve is defined with lines and arcs. This element is composed of several line, arc or point elements. The curve element represents contour geometry that consists of line and arc segments. Consecutive line or arc segments should be connected to each other. Starting with the point element and using the end point of previous field for the starting point of the next line or arc element removes data redundancy. This element can implement both closed and open contours depending on element requirements.

Content
CONTENT DECLARATION: ((point|line|arc), (line|arc)*)

- arc—Arc geometry
- line—Line geometry
- point—Point definition

Attributes
None

Parent Elements

- operation-contour—A contour following machining operation
- operation-pocket—A pocket cutting operation
- tool—A cutting tool definition

♦ postal-code

ZIP Code

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements

- address—Part designer's address

♦ profile

Profile tolerance. The value limits the deviation of the curved line of a curved surface for example, walls generated by a POCKET or a CONTOUR). The section of the curved surface should not exceed the corresponding curve tolerance zone.

Content
Empty

Attributes
✓ *value*
This field value range from negative to positive input values.
Type: double, Value(s): CDATA, Default Value: #REQUIRED

Parent Elements

- tolerance-set—A set of actual tolerance ranges

♦ region

State name or major District name

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements

- address—Part designer's address

◆ revision

Part number and Part Revision information

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements

- part—Simple part description

◆ roundness

Roundness tolerance. The tolerance determines the range in which the cross-section of a hole should remain. This value constructs two concentric circles for a specific hole feature where the hole resides. The difference of radius of two concentric circles is determined by this ROUNDNESS element.

Content
Empty

Attributes
✓ *value*
This field value ranges from negative to positive input values.
Type: double, Value(s): CDATA, Default Value: #REQUIRED

Parent Elements

- tolerance-set—A set of actual tolerance ranges

◆ setup

Machining setup and machining operations defined in the setup element. A sub-element "axis" has setup orientation information. The setup element may have more than one pattern element which is used for grouping machining operations. The tolerance-set information may be inherited from the body element or defined in this element with the USE_TOLERANCE attribute.

Content
CONTENT DECLARATION: (axis, (pattern-hole|pattern-contour|pattern-pocket)*)

- axis—The local axis system(LCS)
- pattern-contour—The grouping element which includes operation-contour elements.
- pattern-hole—The grouping element which includes operation-hole elements
- pattern-pocket—The grouping element which includes operation-pocket elements

Attributes
✓ *seq-no*
Value(s): ID, Default Value: #REQUIRED
✓ *use-tolerance*
Value(s): IDREF, Default Value: TOL_-1

Parent Elements

- body—The structural element which includes ordered machining setups

◆ street

Street, Town names

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements
- address—Part designer's address

◆ structure

This element just shows the skeleton of the BODY element. The structure element is designed to help users figure out the process planning in the body element. The structure element provides an indented list, which is composed of symbolic words that stand for each structural element, defined in the body element. Structure is optional, so, this element may appear in the top level of the file.

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements
m-part—The root element of any NCML document

◆ tel

Telephone Number - Simple data element

Content
CONTENT DECLARATION: (#PCDATA)

Attributes
None

Parent Elements
- address—Part designer's address

◆ tolerance-lib

Grouping element of tolerance-set elements. The tolerance-lib element should include at least one tolerance-set. The tolerance-lib element is one of the top-level elements consisting of a NCML document. The tolerance-lib element (structure) must appear in the root (m-part) element.

Content
CONTENT DECLARATION: (tolerance-set*)
- tolerance-set—A set of actual tolerance ranges

Attributes
None

Parent Elements
- m-part—The root element of any NCML documents

◆ tolerance-set

A set of actual tolerance ranges which can be applied for any NCML structural element. This tolerance-set element includes actual values governing real tolerance ranges. With the different values of the type attribute, this tolerance-set may be used for general sizing tolerance or the tolerance that determine the relative error among features. The specific values can be stored contents elements.

Content
CONTENT DECLARATION: (bilateral?,flatness?,roundness?,profile?,concentricity?)
- bilateral—Bilateral sizing tolerance range
- concentricity—Concentricity tolerance range
- flatness—Flatness tolerance range
- profile—Profile tolerance range
- roundness—Roundness tolerance range

Attributes

✓ *index*

Tolerance-set Identification. This should be different from other ID type attributes. This value is referred by use-tolerance attribute in any machining feature element.

Value(s): ID, Default Value: #REQUIRED

✓ *name*

The description of this tolerance-set. This attribute can help users to understand the purpose of the tolerance-set.

Value(s): CDATA, Default Value: temp_tolerance

✓ *type*

Determines the tolerance-set type. This can have one of GENERAL or GROUP. In the case of GENERAL, other field values are applied along the hierarchy from the level that has the index value at the use-tolerance attribute. Otherwise, the tolerance ranges defined in the tolerance-set are only used to define correlated tolerances among the geometry in the features, which has the same index value at the use-tolerance attribute.

Value(s):

- GENERAL
- GROUP

Default Value: GENERAL

Parent Elements

- tolerance-lib—An element which groups "tolerance-sets"

♦ tool

Cutting Tool. The tool element purpose is to deliver cutting tool geometry. Most fields parameterize geometric characteristics. Only type attribute and diameter elements are shared general components, though others are specific to the type attribute value. The curve element should be included in case the type attribute is set to "CUSTM". In this case, the half shape of tool geometry should be expressed in the curve element in the positive XZ plane. The usage of the TOOL element will be explained with examples as follows.

Content

CONTENT DECLARATION:

(diameter,flute-length?,corner-radius?,pitch-width?,h-limit?, point-angle?,apt-parameter?,curve?,is-cutter?)

- apt-parameter—APT 7 parameters: standard tool description
- corner-radius—Cutting tool corner radius
- diameter—Cutting tool diameter
- flute-length—Cutting tool flute length
- h-limit—The h-limit of the tapping tool
- is-cutter—The cutting tool blade information
- pitch-width—The pitch length of the tapping tool
- point-angle—The tool tip angle of the cutting tool
- curve—Curve geometry which consists of lines and arcs

Attributes

✓ *index*

The cutting tool ID. This ID should be different from any IDs in the same NCML document. This index value can be refereed at the use-tool attribute in any pattern element.

Value(s): ID, Default Value: #REQUIRED

✓ *type*

Cutting Tool type. This attribute specifies the cutting tool shape used in the 3-Axis CNC milling operation. The type can have one of the cutting tool types as follows;

CUSTM: the custom tool type in which the geometry consists of consecutive lines and arcs.

APT7P: cutting tool which is defined by APT 7 parameters

DRILL: Drill Machining tool

CTDRL: Center Drill
SPDRL: Spot Drill
CBORE: Counter Bore
CSINK: Counter Sink
REAMT: Reaming Tool
CREAM: Center Reaming Tool
KNURL: Knurling Tool
TAPPG: Tapping Tool
SIDEM: Side End Mill
FLATM: Flat End Mill
BALLM: Ball End Mill
CNRRM: Corner radius end Mill
RNDEM: Round end mill
TPREM: Tapered end mill

Value(s): CUSTM/APT7P/DRILL/CTDRL/SPDRL/CBORE/CSINK/REAMT/CREAM/KNURL/TAPPG/SIDEM/F/ATM/BALLM/CNRRM/RNDEM/TPREM
Default Value: #REQUIRED

Parent Elements

- tool-set—An element which groups tool elements

◆ tool-set

Tool-set is a set of tool elements. The tool-set element contains tool elements, which include description of a cutting tool that may be used in the NCML document. This element is one of the top-level elements consisting of a NCML document and composed of several TOOL elements. The tool-set element (structure) must appear in the root (m-part) element.

Content
CONTENT DECLARATION: (tool+)

- tool—A cutting tool definition

Attributes
None

Parent Elements

- m-part—The root element of any NCML documents

◆ workpiece

This WORKPIECE element describes geometric information of the raw material. Depending on the assigned value of TYPE field, each data field has different meanings. To date, only cube and cylinder shape can be supported.

Content
CONTENT DECLARATION: (material, (cube|cylinder)?)

- cube—Cubic solid geometry
- cylinder—Cylinder solid geometry
- material—The part material type

Attributes
✓ *source*
The file name that includes original work piece geometry. This is only valid if TYPE is set to "FILE". This should be able to support the URI or URL. The file format may be inferred from the file.
Value(s): CDATA Default Value: #IMPLIED
✓ *type*
The type of the work piece shape. Only three options are available.
"seq-no" is composed of two parts which are separated by ":". The first part is a description of this element

and the second part is a number that indicates the order of machining operations. Setup, pattern, or operation with a lower number means that this should be machined before those with higher numbers.
Value(s): CUBE//CYLINDER/FILE, Default Value: #REQUIRED

Parent Elements

- m-part—The root element of any NCML documents

APPENDIX C

APPLICATION ARCHITECTURE

❑ Programming Classes in the FACILE software system

The FACILE system was created using Visual C++, by Microsoft [MS01]. This means that software systems are modeled with C++ classes. For a better understanding of developed software systems in OOP, analyzing class structure is very helpful. Since it is almost impossible to provide all the detailed explanations about programming codes, the organization of the C++ classes in FACILE are given. However, the class organizations corresponding to the NCML hierarchical structure and geometric elements are identical to the elements of NCML provided in Figure 4.2 and the classes only related to implement GUI systems are of less concern. Therefore, the classes related to the FACILE functions related to the output file generation are mainly given in Figure C.1.

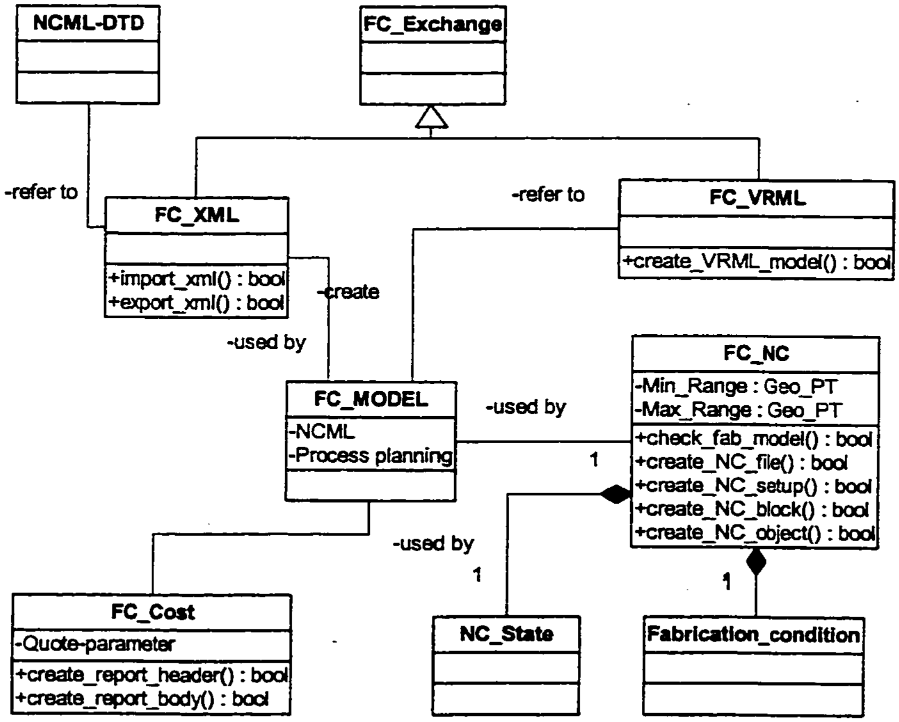


Figure C.1 Simplified FACILE class diagram

In addition to the information in the NCML file, the FACILE internal model (FC_MODEL in Figure C.1) includes process planning information. The surrounding classes use this FACILE internal model to generate output files or information. For example, FC_XML imports or exports an NCML document from the FACILE internal model, while FC_VRML creates a VRML model from NCML information, and FC_Cost estimates a quote for the part described in NCML. Lastly FC_NC generates G-code files corresponding to model, setup, pattern and operation respectively.

☐ FACILE Menu Structure

Figure C.2 shows the menu structure of the currently implemented FACILE software systems. Though the depth of the hierarchical menu structure is three-levels deep in most cases, only two levels are shown in Figure C.2 because of space limitations. The menu items also can be classified by the functional distinction into three groups like functions belonging to the designer (FACILE/Design), the fabricator (FACILE/Fabricate) or the shared functions.

The descriptions of the first level menu items are as follows:

- File: used to manage files; creating new ones, saving files, importing/exporting XML(NCML) files, XML files, printing the display area, and exiting the FACILE program.
- Edit: edit geometry that has already been drawn.
- Draw: contains commands to draw 2D geometry that can then be made into 3D features, also sets pen color, grid and snap.
- Render: used to create hidden line and surface shaded views of the workpiece.
- View: used to modify the view of the workpiece, including zoom functions, face views, and isometric views.
- Window: used to change the window environment and background color.
- Tools: accesses the different tools in FACILE, including Macro Simulation, Quotation Helper, and Configuration options.
- Design: used to create, modify, and delete manufacturing features.
- Fabricate: used to generate NC programs and assign fabrication attributes. Includes Tool Management

and Tool Turret management.

- Help: Help files. Currently, there are no help files for FACILE.

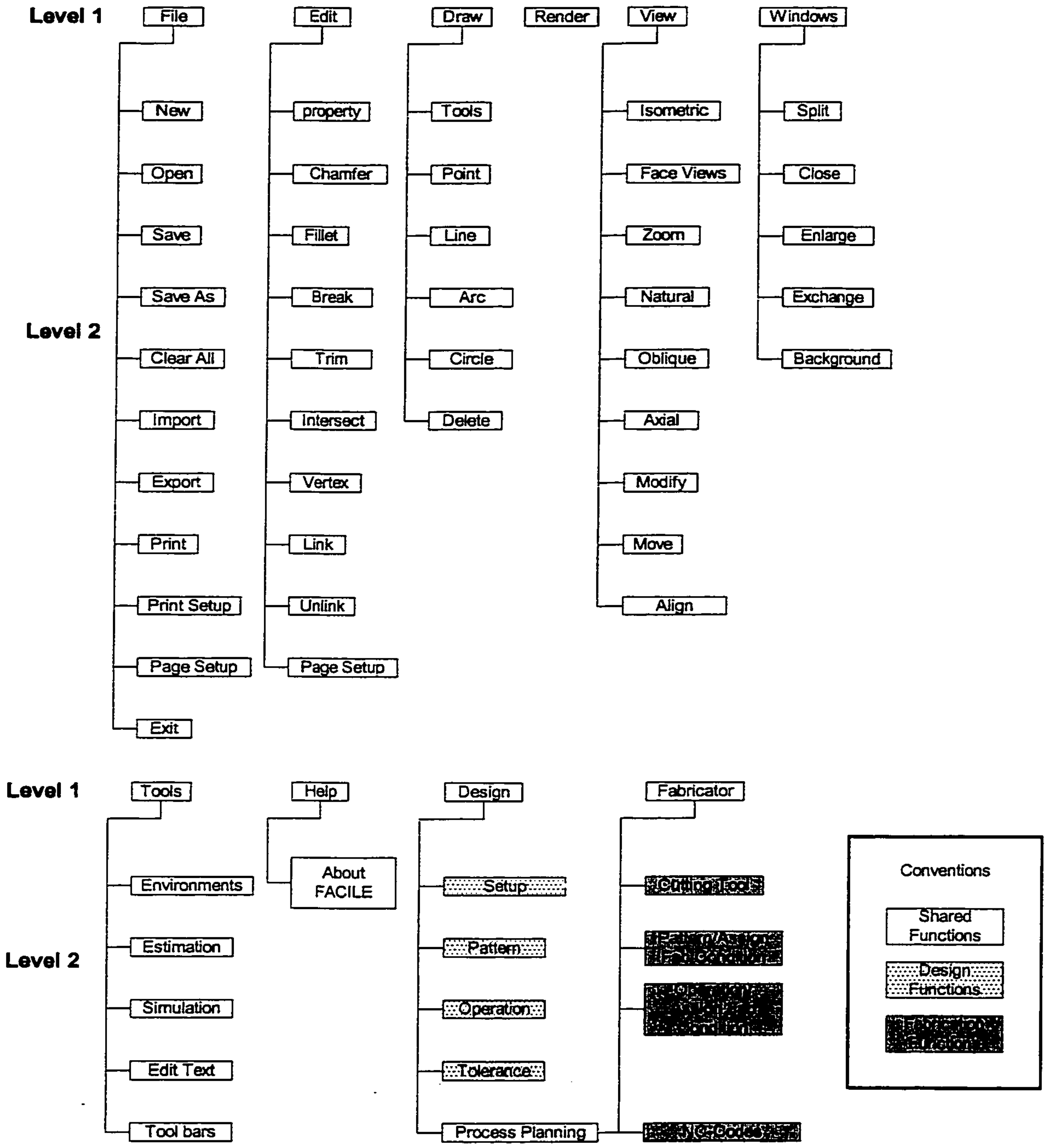


Figure C.2 FACILE software menu structure

❑ E-Mill Database Architecture

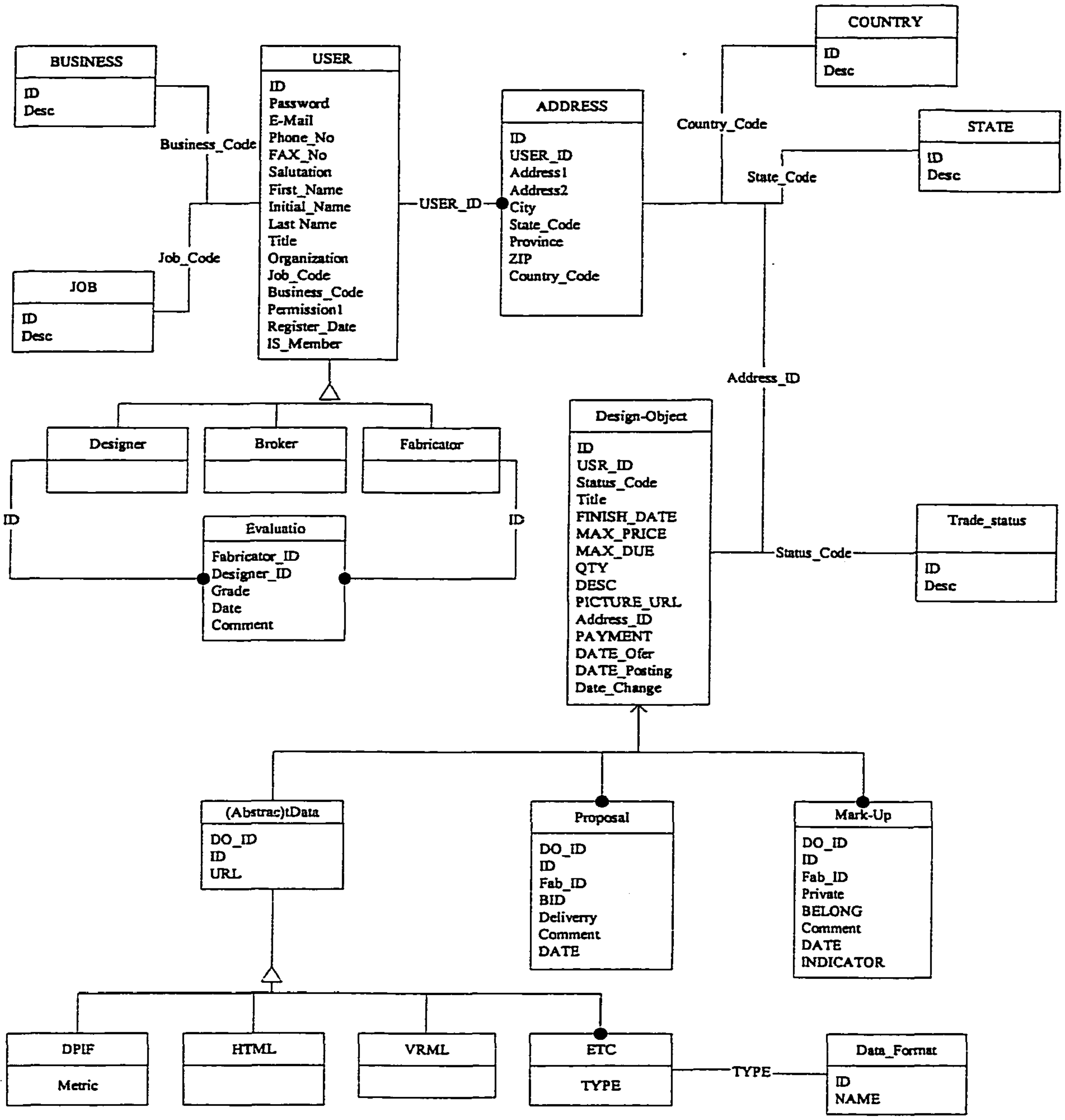


Figure C.3 E-Mill database architecture

Figure C.3 is a diagram showing the database design of E-Mill. Each box shape represents a database table. The table name is shown in the upper compartment of the box. In the lower compartment, attributes belonging to the table are listed. To express the relationship between tables, the attribute linking two tables is placed on the line connecting two tables.

❑ FACILE/Fabricate Data structures

As seen in Figure 5.14, five data libraries are used for executing FACILE/Fabricate functions. The contents of each library are given here. Each library includes multiple sets of the related information and each set has several attributes. Each data structure is implemented as a C++ class, so the member variables in the C++ class are used with its real name and data type in programming codes.

Table C.1 Machine tool library.

Name	Descriptions
d_max_x	Workable workpiece size along the x axis
d_max_y	Workable workpiece size along the y axis
d_max_z	Workable workpiece size along the z axis
d_min_tol	Achievable tolerance value for the specific CNC machining

*The machine tool library has not been implemented yet in the current software systems.

Table C.2 Machining (Cutting) conditions library.

Name	Descriptions
	Attributes for G-codes generation
d_clearance	The z level of the machining clearance plane
d_def_tool_length	Default Tool length used for cutting simulation.
i_block_increment	Incremental amount of G-codes block number
i_block_start	Start number of G-codes block number
b_use_collant	Determine the coolant usage.
	Attributes for center drilling
d_depth_center_drill	Default cutting depth of center drilling
d_spindle_center_drill	Default spindle speed of center drilling
d_feed_center_drill	Default federate of center drilling
	Attributes for drilling
i_drilling_cycle	Drilling cycle Mode. G80, G81, G82, G83, etc
d_spindle_drilling	Default spindle speed of drilling
d_height_retraction	Cutting tool retraction height used for drilling
d_feed_drilling	Default federate of drilling
d_ratio_PECK_Radius	A rate of Peck motion depth over the cutting tool radius.
	Attributes for the contour operation
b_rough_con	A flag if roughing is to be used in the contour feature
b_s_finish_con	A flag if semi-finishing is to be used in the contour feature
b_finish_con	A flag if finishing is to be used in the contour feature
i_retract_method	The cutter retraction method in generating g-codes for contouring
i_approach_method	The cutter approach method in generating g-codes for contouring
d_rough_allowance	Material thickness left after contour rough cutting

d_s_finish_allowance	Material thickness left after contour semi-finish cutting
d_finish_allowance	Material thickness left after contour finish cutting
	Attributes for pocketing
i_pocket_method	Default pocketing method like spiral-out, spiral-in or parallel cutting
bp_rough_con	A flag if roughing is to be used in the pocket feature
bp_s_finish_con	A flag if semi-finishing is to be used in the pocket feature
bp_finish_con	A flag if finishing is to be used in the pocket feature
ip_approach_method	The cutter retraction method in generating g-codes for pocketing
dp_rough_allowance	Material thickness left after pocket rough cutting
dp_s_finish_allowance	Material thickness left after pocket semi-finish cutting
dp_finish_allowance	Material thickness left after pocket finish cutting
	Attributes for cutting tool approach and retraction
d_angle_approach	Cutting Tool approach angle
d_height_approach	Cutting tool approach height
d_angle_arc_approach	Arc angle used for the arc approach method
d_diam_arc_approach	Arc Diameter for the arc approach method
d_dist_linear_approach	The approach length of tool path for the linear approach
	Milling General
d_depth_path_Rough	Ratio of cutting depth over cutter diameter for pocketing
d_dist_path_Rough	Ratio of adjacent tool path over cutter diameter for pocketing
d_feed_finish	The default feed rate for finish cutting
d_feed_sfinish	The default feed rate for semi-finish cutting
d_feed_rough	The default feed rate for rough cutting
i_spindle_finish	Default spindle speed for finish cutting
i_spindle_rough	Default spindle speed for semi-finish cutting
i_spindle_sfinish	Default spindle speed for rough cutting

Table C.3 Tool turret library.

Name	Descriptions
i_position	Tool position number in the tool turret
cs_name	Cutting tool name
i_type	Cutting tool type. Refer to Appendix B.4 for the defined tool type
i_source	Cutting tool source. Source tool library name
d_radius	Cutter radius

Quote parameter library

All quote parameter are defined in Table D.2.

Cutting Tool Library

The contents of any cutting tool are identical to those of the tool element in the NCML DTD. Users are referred to Appendix B.4 for the detailed specification of tool elements.

APPENDIX D

QUOTING PROCEDURE

☐ Parameter calibration for Quote Estimation

Tables and figures in this appendix illustrate the procedure of the quoting parameter calibration.

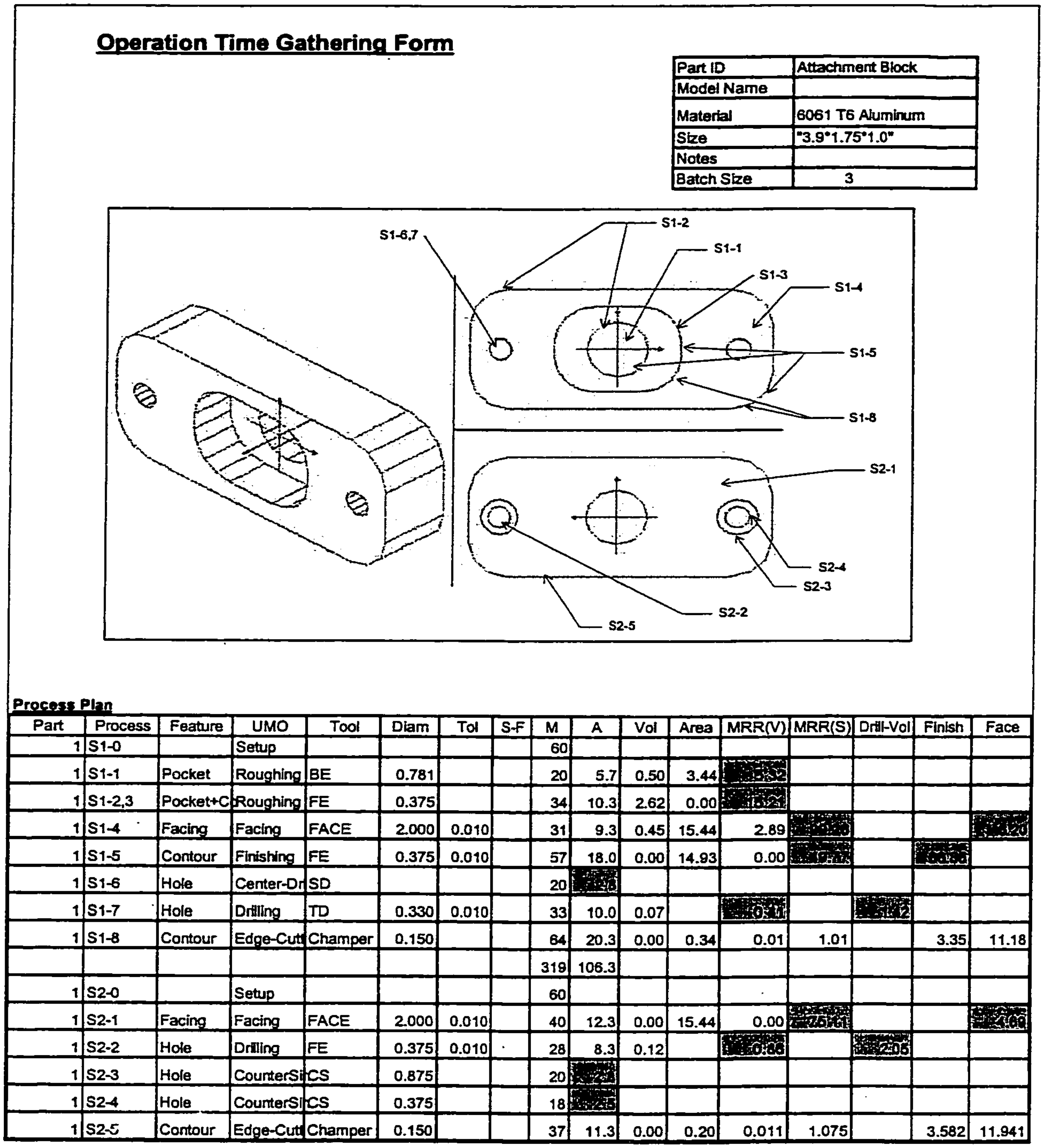
Data gathering and calculation

The sheet in Figure D.1 was used to get cutting parameters related to the quotation calibration at the Stone Machine Company. It mainly keeps a record of machining operation times for machined parts.

The data-gathering sheet is composed of three parts. On the top left, there is a table describing general information on the machined parts. It includes part name, material and machining batch size. In the middle, a picture shows three different views of the object parts. Each process name is noted with an arrow. This process name matches the content in the source row of the process plan table at the bottom. While this data gathering process seems laborious, it could certainly be automated by employing an OAC run by an ordinary PC.

To understand the process plan table (Figure D.1), descriptions of the header row contents are given as follows:

- Diam: Cutter Diameter, OD
- Tol: Tolerance value assigned on the target operation
- S-F: Surface finishing level assigned on the target operation.
- M: Machining times of each operation
- A: Average machining time. M/batch size.
- Vol: Volume removed from the machining operation
- Area: Area swept from the machining operation
- MRR(V): Volumetric Material removal rate=Vol/A
- MRR(S): Area cutting rate=Area/A
- Drill-Vol: Normalized drilling volume rate. See equation (5.7)
- Face: Normalized facing area rate. See equation (5.8)
- Finish: Normalized surface finish rate. See equation (5.9)



Operation Time Gathering Form

Part ID	Attachment Block
Model Name	
Material	6061 T6 Aluminum
Size	"3.9*1.75*1.0"
Notes	
Batch Size	3

Process Plan

Part	Process	Feature	UMO	Tool	Diam	Tol	S-F	M	A	Vol	Area	MRR(V)	MRR(S)	Drill-Vol	Finish	Face
1	S1-0		Setup					60								
1	S1-1	Pocket	Roughing	BE	0.781			20	5.7	0.50	3.44	[illegible]				
1	S1-2,3	Pocket+C	Roughing	FE	0.375			34	10.3	2.62	0.00	[illegible]				
1	S1-4	Facing	Facing	FACE	2.000	0.010		31	9.3	0.45	15.44	2.89	[illegible]			[illegible]
1	S1-5	Contour	Finishing	FE	0.375	0.010		57	18.0	0.00	14.93	0.00	[illegible]		[illegible]	
1	S1-6	Hole	Center-Dr	SD				20	[illegible]							
1	S1-7	Hole	Drilling	TD	0.330	0.010		33	10.0	0.07		[illegible]		[illegible]		
1	S1-8	Contour	Edge-Cut	Champer	0.150			64	20.3	0.00	0.34	0.01	1.01		3.35	11.18
								319	106.3							
1	S2-0		Setup					60								
1	S2-1	Facing	Facing	FACE	2.000	0.010		40	12.3	0.00	15.44	0.00	[illegible]			[illegible]
1	S2-2	Hole	Drilling	FE	0.375	0.010		28	8.3	0.12		[illegible]		[illegible]		
1	S2-3	Hole	CounterSi	CS	0.875			20	[illegible]							
1	S2-4	Hole	CounterSi	CS	0.375			18	[illegible]							
1	S2-5	Contour	Edge-Cut	Champer	0.150			37	11.3	0.00	0.20	0.011	1.075		3.582	11.941

Figure D.1 Sample cutting time gathering sheet.

Table D.1 Organized cutting time information gathered for calibrating quoting parameters.

Part	UMO	Diam	Tol	S-F	M	A	Vol	Area	RR(V	RR(S	Drill-Vol	Finish	Face	Avg	
1	Center-Drilling				20.00	[illegible]									
3	Center-Drilling	0.500		125	64.00	[illegible]									
2	Center-Drilling*	0.500			23.00	[illegible]									Discard
1	CounterSink	0.875			20.00	[illegible]									
1	CounterSink	0.375			18.00	[illegible]								2.77	
1	Drilling	0.330	0.010		33.00	10.00	0.07		[illegible]		[illegible]				
1	Drilling	0.375	0.010		28.00	8.33	0.12		[illegible]		[illegible]			1.73	
2	Drilling	0.332	0.005		39.00	56.00	0.12		[illegible]		[illegible]				
2	Drilling	0.438	0.005		223.00	220.00	0.06		[illegible]		[illegible]				Discard
2	Drilling	0.120	0.005		20.00	17.00	0.00		[illegible]		[illegible]				Discard
2	Drilling	0.125	0.005		17.00	14.00	0.00		[illegible]		[illegible]				Discard
3	Drilling	0.484	0.005	125	46.00	46.00	0.66		[illegible]		[illegible]				
3	Drilling	0.281	0.005	125	23.00	23.00	0.09		[illegible]		[illegible]				
3	Drilling	0.313	0.005	125	22.00	22.00	0.05		[illegible]		[illegible]				
3	Drilling	0.200	0.005	125	28.00	28.00	0.03		[illegible]		[illegible]				
3	Drilling	0.250	0.005	125	39.00	39.00	0.14		[illegible]		[illegible]			1.00	
2	Edge-Cutting	1.000	0.002		31.00	28.00	0.04	0.44	0.09	0.94		0.47			
1	Edge-Cutting	0.150			64.00	20.33	0.00	0.34	0.01	1.01		3.35			
1	Edge-Cutting	0.150			37.00	11.33	0.00	0.20	0.01	1.07		3.58			
1	Facing	2.000	0.010		31.00	9.33	0.45	15.44	2.89	[illegible]			[illegible]		
1	Facing	2.000	0.010		40.00	12.33	0.00	15.44	0.00	[illegible]			[illegible]	5.45	
2	Facing	3.000	0.005		75.00	9.00	0.92	11.56	6.13	[illegible]			[illegible]		
2	Facing	3.000	0.005		32.00	14.50	4.80	11.56	19.86	[illegible]			[illegible]	1.74	
3	Finish-Boring	0.875	0.001	8	304.00	301.00		7.70	0.00	[illegible]		[illegible]			
3	Finish-Boring	1.125	0.001	8	161.00	158.00		2.47	0.00	[illegible]		[illegible]		0.65	
1	Finishing	0.375	0.010		57.00	18.00	0.00	14.93	0.00	[illegible]		[illegible]		66.36	
2	Finishing	0.375	0.005		255.00	31.50	0.00	8.50	0.00	[illegible]		[illegible]			
3	Finishing	0.500	0.005	125	144.00	70.50	5.88	9.35	5.00	[illegible]		[illegible]			
3	Finishing	0.500	0.005	125	144.00	141.00	5.88	9.35	2.50	[illegible]		[illegible]		11.17	
2	Finishing	0.375	0.002		102.00	99.00	0.19	3.05	0.12	[illegible]		[illegible]			
2	Finishing	0.500	0.002		67.00	64.00	0.07	1.71	0.07	[illegible]		[illegible]			
2	Finishing	0.375	0.002		102.00	99.00	0.19	3.05	0.12	1.85		[illegible]			
2	Finishing	0.500	0.002		67.00	64.00	0.07	1.71	0.07	1.61		[illegible]		2.04	
3	Roughing	0.500	0.005	125	144.00	70.50	5.88	9.35	[illegible]						
1	Roughing	0.781			20.00	5.67	0.50	3.44	[illegible]						
1	Roughing	0.375			34.00	10.33	2.62	0.00	[illegible]						
2	Roughing	0.500			200.00	24.63	2.03		[illegible]					7.62	
1	Setup				60.00										
1	Setup				60.00										
2	Setup				150.00										
2	Setup				60.00										
2	Setup				60.00										
3	Setup				180.00									95.00	
3	Tapping	0.200	0.005		19.00	16.00	0.03	0.88	0.11	[illegible]		[illegible]			Discard
3	Tapping	0.313	0.005		12.00	9.00	0.05	0.83	0.35	[illegible]		[illegible]			Discard
3	Tapping	0.250	0.005		31.00	28.00	0.14	3.30	0.30	[illegible]		[illegible]			Discard
3		0.375	0.005	125	203.00	200.00			0.00	0.00		0.00			Discard

* The values in the "Ave(Average) column is used for determining quoting parameters.

Determine Quoting parameters

Time data related to machining operations is gathered. This information is sorted by UMO, tolerance and surface finish in the next table. Normalized MRRs are calculated and associated with their respective UMO types, tolerance value and surface finish imposed on the UMO. These normalized MRRs are shown in the shaded box in Table D.1 – each row has either one or two shaded boxes. Basically we are using the average of these MRRs for the quotation helper calibration. The average values for each MRR and other time factors are shown in the "AVG" column.

The Table D.2 shows how the MRR ratio changes with regard to different tolerances and surface finishes. Based on the average MRR attained from the Table D.1, the ratio of different tolerances and surface finishes is calculated by simple arithmetic. The shaded row contains the default MRRs and the shaded column includes various ratios depending on the different tolerances and surface finish levels. These values were used in the quotation comparison test in section 7.2.2.

Table D.2 Quoting parameter calculated.

		Roughing	Finishing	Ratio	Drilling	Ratio	Facing	Ratio	Weight Factor
Default		7.619	66.356		1.734		5.449		
Tolerance	0.01		66.356	1.000	1.734	1.000	5.449	1.000	[illegible]
	0.005		14.770	0.223	1.000	0.577	1.735	0.318	0.573
	0.002		2.037	0.031					[illegible]
	0.0005		0.647	0.010					0.010
Surface Finish									
	125		14.770	1.000					[illegible]
	8		0.647	0.044					[illegible]

❑ Quoting procedures used in 7.2.3

Quoting procedures are carried out with the part used for the quoting range test in 7.2.3. Besides estimated machining time which can be attained from FACILE/Fabricate, other cost sensitive factors need to be considered. These cost factors were regarded as fixed values for the convenience and consistency of the estimated quote. To calculate quotes follow the cost model presented in 5.3.2. The same symbolic variables are used in the following tables. With simple modifications of the cost model, we can simplify the quoting calculation procedure. As seen in Table D.3, from the data given from Stone Machine Company, we achieved three fixed values for quoting calculation. These values are shaded in the tables.

Table D.3 Fixed cost factors used in the quoting process

Cost items	Cost/Hour($/h)	Hours/Part	Cost Each($)	Hour / Order	Cost/Order
Material*			0		
Painting/Plating			1		
Deburring/Finishing	45	0.01667	0.75		
Total Fixed Cost per Part			1.75		
Programming	60			2	120
SetUp	60			2	120
Run Time(Cm)	60				
Final Inspection/Labeling	45			0.5	22.5
Total Fixed Cost per Order					262.5

Note: In this particular case all parts were specified as "material provided by the buyer". Therefore, material cost is omitted.

The cost model equation (5.1) can be rewritten as follows:

$$C{=}C_m\,(T_m + T_p) + Q\,(C_w + C_p) = C{=}C_m\,T_m + C_{fix\_o} + Q\,C_{fix\_p} \qquad \text{(D.1)}$$

Where,

C_{fix_o}: Total Fixed Cost per order (\$262.5)

$C_{fix\_p:}$ Total Fixed Cost per Part (\$1.75)

Since $C_m\,T_m$ is the output of FACILE Quotation helper, all cost factors can be determined.

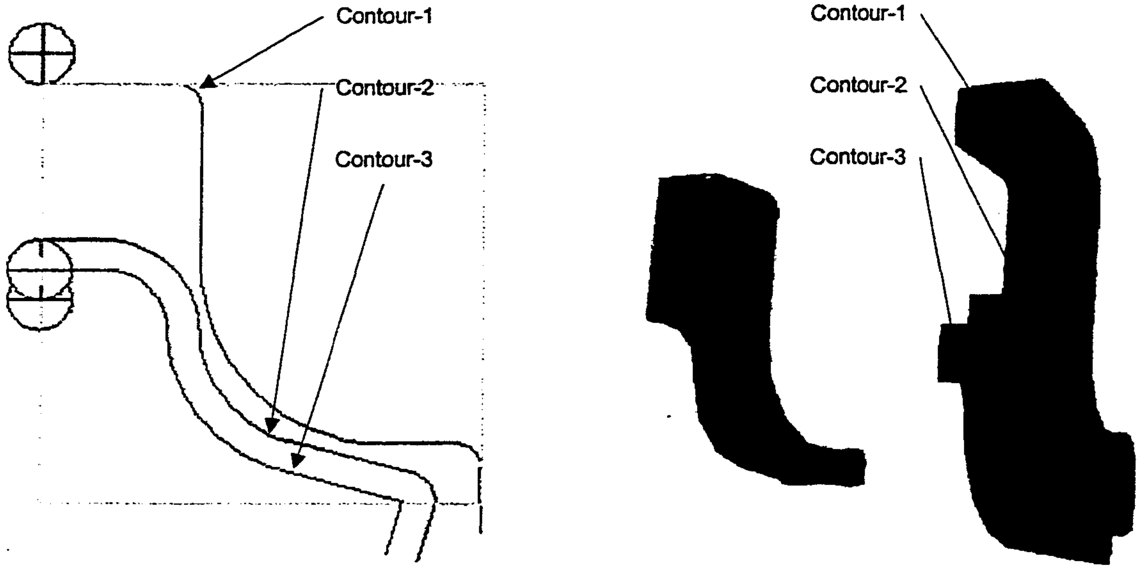
The quoting procedures for test parts are given in the following tables.

Table D.4 Quotes generated from FACILE/Fabricate for the target parts

Part ID	RFQ-884	RFQ-1081	RFQ-1093	RFQ-953	RFQ-1113A	RFQ-1113B
Quantity (Q)	250	25	1	35	10	10
Fixed cost/order ($C_{fix\ o}$)	262.5	262.5	262.5	262.5	262.5	262.5
Q $C_{fix\ p}$	437.5	43.75	1.75	61.25	17.5	17.5
Cutting Cost ($C_m\,T_m$)	619	653.24	9.64	1134.31	23.5	29.7
Quote Estimated(C)	1319	959.49	273.89	1458.06	303.5	309.7
Unit cost	5.276	38.38	273.89	41.65	30.35	30.97
($C_m\,T_m$)/C	47	68	4	78	8	10

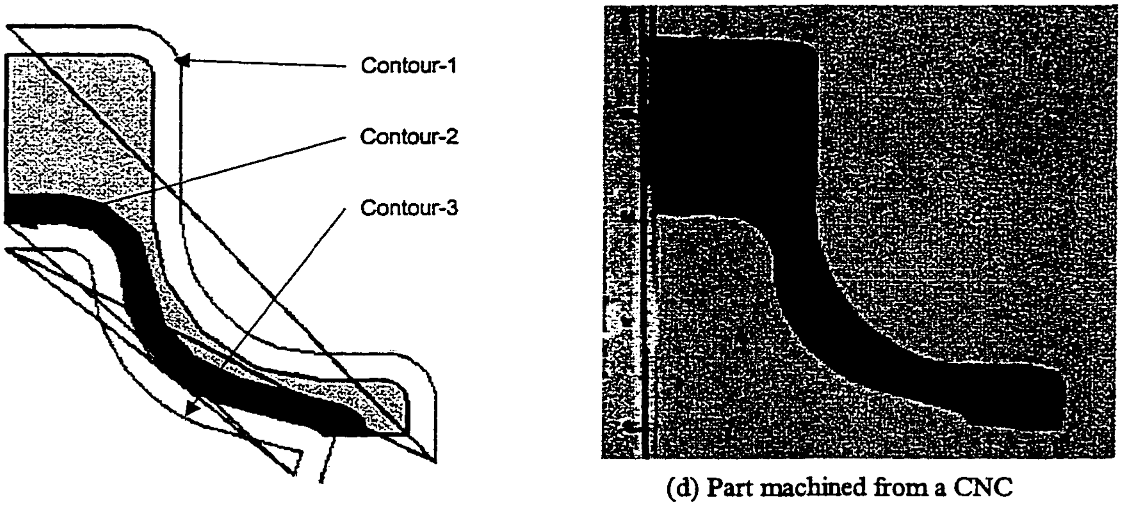
APPENDIX E

PART SAMPLE



(a) Graphical NCML representation. This part is composed of three operation-contour belonginhg to one pattern-contour element.

(b) NCML VRML representation



(c) G-codes on the (Macro) simulated part.

(d) Part machined from a CNC machine with the generated G-codes

*Part design source: The National Design Repository at Drexel University

Figure E.1 Diverse part formats generated in FACILE.

A full set of data generated for a specific part through the NCML machined part fabrication process is provided. This includes NCML, VRML and G-codes files generated during the fabricating process. The whole document may help readers gain a better understanding of the overall process. Since the file size is fairly large, a simple shaped part was chosen as an example. Comments corresponding to NCML elements are inserted for a better understanding between lines in the code.

NCML File

```
<?xml version='1.0' encoding='ISO-8859-1' ?>
<!DOCTYPE m-part SYSTEM "http://cadcam3.sr.unh.edu/dtd/NCML.dtd">
<m-part unit = "INCHES" version = "DPIF(FACILE Version 1.0)">
    <header>
        <part-spec>
            <name/>
            <date/>
            <revision>Blank-Part Number</revision>
        </part-spec>
        <address>
            <name>JOHN DOE</name>
            <tel>603)862-1234</tel>
            <email>JOHN.DOE@UNH.EDU</email>
        </address>
    </header>
    <workpiece type = "CUBE">
        <material>BLANK :Default Material</material>
        <cube>
            <point z = "-0.740000" use = "Minimum Point"/>
            <point x = "3.500000" y = "3.250000" use = "Maximum Point"/>
        </cube>
    </workpiece>
    <body>
        <setup seq-no = "S_1">
            <axis>
                <point use = "ORIGIN"/>
                <point z = "1.000000" use = "Z-Axis"/>
                <point x = "1.000000" use = "X-Axis"/>
            </axis>
            <pattern-contour seq-no = "S_1-PC_1" use-tool = "TOOL_1">
                <operation-contour seq-no = "S_1-PC_1-OP_1">
                    <depth value = "0.750000"/>
                    <offset direction = "LEFT"/>
                    <curve>
                        <line>
                            <point x = "3.500000" y = "3.500000"/>
                            <point x = "3.500000" y = "2.450000"/>
                        </line>
                        <arc nvz = "1.000000" direction = "CW">
                            <point x = "3.300000" y = "2.450000"/>
                            <point x = "3.300000" y = "2.250000"/>
                        </arc>
                        <line>
```

```
                        <point x = "2.000000" y = "2.250000"/></line>
                <arc nvz = "1.000000" direction = "CCW">
                        <point x = "2.000000" y = "0.750000"/>
                        <point x = "0.500000" y = "0.750000"/>
                </arc>
                <line>
                        <point x = "0.500000" y = "0.200000"/></line>
                <arc nvz = "1.000000" direction = "CW">
                        <point x = "0.300000" y = "0.200000"/>
                        <point x = "0.300000"/>
                </arc>
                <line>
                        <point x = "-0.250000"/></line>
        </curve>
</operation-contour>
<operation-contour seq-no = "S_1-PC_1-OP_2">
        <depth value = "0.750000"/>
        <offset direction = "RIGHT"/>
        <curve>
                <line>
                        <point x = "1.950000" y = "3.500000"/>
                        <point x = "1.950000" y = "3.007179"/>
                </line>
                <arc nvz = "1.000000" direction = "CW">
                        <point x = "1.450000" y = "3.007179"/>
                        <point x = "1.479361" y = "2.508042"/>
                </arc>
                <line>
                        <point x = "1.409478" y = "2.503931"/></line>
                <arc nvz = "1.000000" direction = "CCW">
                        <point x = "1.482881" y = "1.256087"/>
                        <point x = "0.273359" y = "1.571615"/>
                </arc>
                <line>
                        <point x = "0.022677" y = "0.610668"/></line>
                <line>
                        <point x = "-0.431320" y = "0.740381"/></line>
        </curve>
</operation-contour>
<operation-contour seq-no = "S_1-PC_1-OP_3">
        <depth value = "0.500000"/>
        <offset direction = "RIGHT"/>
        <curve>
                <line>
                        <point x = "2.200000" y = "3.500000"/>
                        <point x = "2.200000" y = "3.007179"/>
                </line>
                <arc nvz = "1.000000" direction = "CW">
                        <point x = "1.450000" y = "3.007179"/>
                        <point x = "1.494042" y = "2.258473"/>
                </arc>
                <line>
                        <point x = "1.424159" y = "2.254362"/></line>
                <arc nvz = "1.000000" direction = "CCW">
                        <point x = "1.482882" y = "1.256085"/>
                        <point x = "0.515263" y = "1.508510"/>
```

```
                        </arc>
                        <line>
                            <point x = "0.264581" y = "0.547562"/></line>
                        <arc nvz = "1.000000" direction = "CW">
                            <point x = "0.022677" y = "0.610668"/>
                            <point x = "-0.046003" y = "0.370287"/>
                        </arc>
                        <line>
                            <point x = "-0.500000" y = "0.500000"/></line>
                    </curve>
                </operation-contour>
            </pattern-contour>
        </setup></body>
    <tool-set>
        <tool type = "FLATM" index = "TOOL_1">
            <diameter value = "0.500000"/></tool></tool-set>
    <tolerance-lib/>
</m-part>
```

VRML File

```
#VRML V2.0 utf8 CosmoWorlds V1.0

# Finished part geometry. Basically mutually conneted polygons can construct solid shape in VRML.

Group {
  children [
    Shape {
      appearance Appearance {
        material Material { diffuseColor 0.65 0.65 0.65          }
      }
      geometry IndexedFaceSet {
        coord Coordinate{
          point [
            0.300 0.000 -0.740, 0.000 0.000 0.000, 0.000 0.000 -0.740, 0.300 0.000 0.000,
            3.500 3.250 -0.740, 3.500 2.450 0.000, 3.500 2.450 -0.740, 3.500 3.250 0.000,
            0.400 0.027 0.000, 0.473 0.100 0.000, 0.500 0.200 0.000, 0.500 0.750 0.000,
            0.702 1.880 0.000, 0.614 1.324 0.000, 1.023 2.144 0.000, 0.939 1.811 0.000,
            1.426 2.136 0.000, 2.000 2.250 0.000, 3.377 2.265 0.000, 3.300 2.250 0.000,
            3.441 2.309 0.000, 3.485 2.373 0.000, 2.200 3.007 0.000, 2.200 3.250 0.000,
            2.107 2.645 0.000, 1.850 2.373 0.000, 1.494 2.258 0.000, 1.424 2.254 0.000,
            0.515 1.509 0.000, 0.265 0.548 0.000, 0.221 0.459 0.000, 0.147 0.394 0.000,
            0.053 0.362 0.000, 0.000 0.367 0.000, 0.044 0.769 -0.740, 0.000 0.731 -0.740,
            0.087 0.858 -0.740, 0.273 1.572 -0.740, 0.506 2.036 -0.740, 0.908 2.366 -0.740,
            1.479 2.508 -0.740, 1.409 2.504 -0.740, 1.717 2.584 -0.740, 1.888 2.766 -0.740,
            1.950 3.007 -0.740, 1.950 3.250 -0.740, 3.485 2.373 -0.740, 3.441 2.309 -0.740,
            3.377 2.265 -0.740, 3.300 2.250 -0.740, 2.000 2.250 -0.740, 1.426 2.136 -0.740,
            0.939 1.811 -0.740, 0.614 1.324 -0.740, 0.500 0.750 -0.740, 0.500 0.200 -0.740,
            0.473 0.100 -0.740, 0.400 0.027 -0.740, 0.000 0.367 -0.500, 0.000 0.731 -0.500,
            2.200 3.250 -0.500, 1.950 3.250 -0.500, 1.950 3.007 -0.500, 0.087 0.858 -0.500,
            0.044 0.769 -0.500, 1.888 2.766 -0.500, 1.717 2.584 -0.500, 1.479 2.508 -0.500,
            1.409 2.504 -0.500, 0.908 2.366 -0.500, 0.506 2.036 -0.500, 0.273 1.572 -0.500,
            2.200 3.007 -0.500, 0.053 0.362 -0.500, 0.147 0.394 -0.500, 0.221 0.459 -0.500,
            0.265 0.548 -0.500, 0.515 1.509 -0.500, 0.702 1.880 -0.500, 1.023 2.144 -0.500,
            1.424 2.254 -0.500, 1.494 2.258 -0.500, 1.850 2.373 -0.500, 2.107 2.645 -0.500,
```

```
          ]
        }
        coordIndex[
          0, 1, 2, -1, 1, 0, 3, -1, 4, 5, 6, -1, 5, 4, 7, -1, 8, 1, 3, -1, 1, 8, 9, -1, 1, 9, 10, -1, 1, 10, 11, -1,
          12, 11, 13, -1, 14, 13, 15, -1, 14, 15, 16, -1, 7, 16, 17, -1, 18, 17, 19, -1, 17, 18, 20, -1,
          17, 20, 21, -1, 17, 21, 5, -1, 17, 5, 7, -1, 22, 7, 23, -1, 7, 22, 24, -1, 7, 24, 25, -1,
          16, 25, 26, -1, 16, 26, 27, -1, 16, 27, 14, -1, 13, 14, 12, -1, 11, 12, 28, -1, 11, 28, 29, -1,
          1, 29, 30, -1, 1, 30, 31, -1, 1, 31, 32, -1, 1, 32, 33, -1, 29, 1, 11, -1, 25, 16, 7, -1,
          34, 2, 35, -1, 2, 34, 36, -1, 2, 36, 37, -1, 2, 37, 38, -1, 2, 38, 39, -1, 40, 39, 41, -1,
          6, 40, 42, -1, 4, 42, 43, -1, 4, 43, 44, -1, 4, 44, 45, -1, 42, 4, 6, -1, 39, 6, 46, -1,
          39, 46, 47, -1, 39, 47, 48, -1, 39, 48, 49, -1, 39, 49, 50, -1, 39, 50, 51, -1, 39, 51, 52, -1,
          39, 52, 53, -1, 2, 53, 54, -1, 2, 54, 55, -1, 2, 55, 56, -1, 2, 56, 57, -1, 2, 57, 0, -1,
          53, 2, 39, -1, 6, 39, 40, -1, 58, 1, 33, -1, 35, 58, 59, -1, 58, 35, 2, -1, 58, 2, 1, -1,
          21, 6, 5, -1, 6, 21, 46, -1, 20, 46, 21, -1, 46, 20, 47, -1, 18, 47, 20, -1, 47, 18, 48, -1,
          19, 48, 18, -1, 48, 19, 49, -1, 17, 49, 19, -1, 49, 17, 50, -1, 16, 50, 17, -1, 50, 16, 51, -1,
          15, 51, 16, -1, 51, 15, 52, -1, 13, 52, 15, -1, 52, 13, 53, -1, 11, 53, 13, -1, 53, 11, 54, -1,
          10, 54, 11, -1, 54, 10, 55, -1, 9, 55, 10, -1, 55, 9, 56, -1, 8, 56, 9, -1, 56, 8, 57, -1,
          3, 57, 8, -1, 57, 3, 0, -1, 60, 45, 61, -1, 7, 60, 23, -1, 45, 7, 4, -1, 7, 45, 60, -1, 61, 44, 62, -1,
          44, 61, 45, -1, 63, 34, 64, -1, 34, 63, 36, -1, 44, 65, 62, -1, 65, 44, 43, -1, 43, 66, 65, -1,
          66, 43, 42, -1, 42, 67, 66, -1, 67, 42, 40, -1, 40, 68, 67, -1, 68, 40, 41, -1, 41, 69, 68, -1,
          69, 41, 39, -1, 39, 70, 69, -1, 70, 39, 38, -1, 38, 71, 70, -1, 71, 38, 37, -1, 37, 63, 71, -1,
          63, 37, 36, -1, 34, 59, 64, -1, 59, 34, 35, -1, 61, 72, 60, -1, 72, 61, 62, -1, 72, 62, 65, -1,
          72, 65, 66, -1, 70, 66, 67, -1, 69, 67, 68, -1, 67, 69, 70, -1, 63, 70, 71, -1, 58, 63, 64, -1,
          58, 64, 59, -1, 70, 58, 73, -1, 70, 73, 74, -1, 70, 74, 75, -1, 70, 75, 76, -1, 70, 76, 77, -1,
          70, 77, 78, -1, 70, 78, 79, -1, 66, 79, 80, -1, 72, 80, 81, -1, 72, 81, 82, -1, 72, 82, 83, -1,
          80, 72, 66, -1, 79, 66, 70, -1, 58, 70, 63, -1, 30, 76, 75, -1, 76, 30, 29, -1, 31, 75, 74, -1,
          75, 31, 30, -1, 32, 74, 73, -1, 74, 32, 31, -1, 33, 73, 58, -1, 73, 33, 32, -1, 22, 60, 72, -1,
          60, 22, 23, -1, 83, 22, 72, -1, 22, 83, 24, -1, 82, 24, 83, -1, 24, 82, 25, -1, 81, 25, 82, -1,
          25, 81, 26, -1, 80, 26, 81, -1, 26, 80, 27, -1, 79, 27, 80, -1, 27, 79, 14, -1, 78, 14, 79, -1,
          14, 78, 12, -1, 77, 12, 78, -1, 12, 77, 28, -1, 76, 28, 77, -1, 28, 76, 29, -1,
        ]
      }
    }
  ]
}

# View Point Definition. This view point is corresponding to the setup element in NCML
DEF SET_UP_1 Viewpoint {
  position 1.750 1.625 6.880 orientation 0.000 0.000 1.000 0.000 description "SET_UP 1"  }
USE SET_UP_1

# FACILE Operation STRUCTURE
# Set Up 1
#Contour-1 is described below.
Transform { center 0 0 0  rotation 0.000 0.000 1.000 0.000    translation 0.000 0.000 0.000
  children [
    DEF S1_FC1_O1 Transform {
      children [
        DEF TS_S1_FC1_O1 TouchSensor{}
         Shape {
           appearance Appearance {material Material { diffuseColor 0.0 1.0 0.0 }}
           geometry Extrusion {
             crossSection [0.00 -0.71, -0.24 -0.70, -0.24 0.00, 0.24 0.00, 0.24 -0.70, 0.00 -0.71,]
             spine[ 3.75 3.50 0.00, 3.75 2.45 0.00, 3.72 2.28 0.00, 3.62 2.13 0.00, 3.47 2.03 0.00,
               3.30 2.00 0.00, 2.00 2.00 0.00, 1.52 1.90 0.00, 1.12 1.63 0.00, 0.85 1.23 0.00,
               0.75 0.75 0.00, 0.75 0.20 0.00, 0.72 0.03 0.00, 0.62 -0.12 0.00, 0.47 -0.22 0.00,
```

```
                0.30 -0.25 0.00, -0.25 -0.25 0.00]
              }
            }
          ]
        }
  #Contour-2 is described below.
        DEF S1_FC1_O2 Transform {
          children [
            DEF TS_S1_FC1_O2 TouchSensor{}
            Shape {
              appearance Appearance {material Material { diffuseColor 0.0 1.0 0.0 }}
              geometry Extrusion {
                crossSection [0.00 -0.71,  -0.24 -0.70,  -0.24 0.00, 0.24 0.00, 0.24 -0.70, 0.00 -0.71]
                spine[1.70 3.50 0.00, 1.70 3.01 0.00, 1.68 2.91 0.00, 1.63 2.84 0.00, 1.56 2.78 0.00,
                  1.46 2.76 0.00, 1.39 2.75 0.00, 0.94 2.65 0.00, 0.53 2.42 0.00, 0.22 2.06 0.00,
                  0.03 1.63 0.00, -0.15 0.92 0.00, -0.36 0.98 0.00]
              }
            }
          ]
        }
  #Contour-3 is described below.
        DEF S1_FC1_O3 Transform {
          children [
            DEF TS_S1_FC1_O3 TouchSensor{}
            Shape {
              appearance Appearance {
              material Material { diffuseColor 0.0 1.0 0.0 }
              }
              geometry Extrusion {
                crossSection [  0.00 -0.47, -0.24 -0.47, -0.24 0.00, 0.24 0.00, 0.24 -0.47, 0.00 -0.47  ]
                spine[ 1.95 3.50 0.00, 1.95 3.01 0.00, 1.91 2.82 0.00, 1.81 2.66 0.00, 1.66 2.55 0.00,
                  1.48 2.51 0.00, 1.41 2.50 0.00, 1.03 2.42 0.00, 0.69 2.22 0.00, 0.43 1.93 0.00,
                  0.27 1.57 0.00, 0.02 0.61 0.00, -0.43 0.74 0.00]
              }
            }
          ]
        }
      ]
    }

# FACILE Operation STRUCTURE – Following codes make interactive simulation possible in a VRML
viewer
DEF PULLOUT_SCRIPT Script{
  eventIn  SFBool  fn_S1_FC1_O1
  eventOut SFVec3f val_S1_FC1_O1
  eventIn  SFBool  fn_S1_FC1_O2
  eventOut SFVec3f val_S1_FC1_O2
  eventIn  SFBool  fn_S1_FC1_O3
  eventOut SFVec3f val_S1_FC1_O3
  url "vrmlscript:
    function  fn_S1_FC1_O1(value){
       if(value)
         if(val_S1_FC1_O1[2] != 0) val_S1_FC1_O1 = new SFVec3f(0,0,0);
         else val_S1_FC1_O1 = new SFVec3f(0,0,3.383260); }
    function  fn_S1_FC1_O2(value){
       if(value)
```

```
        if(val_S1_FC1_O2[2] != 0) val_S1_FC1_O2 = new SFVec3f(0,0,0);
        else val_S1_FC1_O2 = new SFVec3f(0,0,3.383260); }
    function fn_S1_FC1_O3(value){
      if(value)
        if(val_S1_FC1_O3[2] != 0) val_S1_FC1_O3 = new SFVec3f(0,0,0);
        else val_S1_FC1_O3 = new SFVec3f(0,0,3.383260); }
    "
}
ROUTE TS_S1_FC1_O1.isActive TO PULLOUT_SCRIPT.fn_S1_FC1_O1
ROUTE PULLOUT_SCRIPT.val_S1_FC1_O1 TO S1_FC1_O1.set_translation
ROUTE TS_S1_FC1_O2.isActive TO PULLOUT_SCRIPT.fn_S1_FC1_O2
ROUTE PULLOUT_SCRIPT.val_S1_FC1_O2 TO S1_FC1_O2.set_translation
ROUTE TS_S1_FC1_O3.isActive TO PULLOUT_SCRIPT.fn_S1_FC1_O3
ROUTE PULLOUT_SCRIPT.val_S1_FC1_O3 TO S1_FC1_O3.set_translation
```

G-Codes

In Figure E.1, G0 movement is drawn with black lines and G1, G2 and G3 are with violet lines.

The g-codes corresponding to the NCML contour operation are indicated in the comments.

```
(Start)
N1G0G17G20G40G90
E0H0
N2G0Z0.5000
N3T0M6H0E1
N4M41
N5S1000M3
N6G0X3.7500Y3.5000
N7Z0.5000
N8Z0.1000
(Contour-1: Step 1)
N9G1Z-0.5000F4.0000
N10Y2.4500F8.0000
N11G2X3.3000Y2.0000I-0.4500
N12G1X2.0000
N13G3X0.7500Y0.7500J-1.2500
N14G1Y0.2000
N15G2X0.3000Y-0.2500I-0.4500
N16G1X-0.2500
N17G0Z0.5000
(Contour-1: Step 2)
N18X3.7500Y3.5000
N19Z-0.4500
N20G1Z-0.7500F4.0000
N21Y2.4500F8.0000
N22G2X3.3000Y2.0000I-0.4500
N23G1X2.0000
N24G3X0.7500Y0.7500J-1.2500
N25G1Y0.2000
N26G2X0.3000Y-0.2500I-0.4500
N27G1X-0.2500
N28G0Z0.5000
N29S1000
N30X1.7000Y3.5000
```

```
N31Z0.1000
(Contour-2: Step 1)
N32G1Z-0.5000F4.0000
N33Y3.0072F8.0000
N34G2X1.4647Y2.7576I-0.2500
N35G1X1.3948Y2.7535
N36G3X0.0315Y1.6347I0.0881J-1.4974
N37G1X-0.1546Y0.9213
N38X-0.3626Y0.9808
N39G0Z0.5000
N40X1.7000Y3.5000
N41Z-0.4500
(Contour-2: Step 2)
N42G1Z-0.7500F4.0000
N43Y3.0072F8.0000
N44G2X1.4647Y2.7576I-0.2500
N45G1X1.3948Y2.7535
N46G3X0.0315Y1.6347I0.0881J-1.4974
N47G1X-0.1546Y0.9213
N48X-0.3626Y0.9808
N49G0Z0.5000
N50S1000
N51X1.9500Y3.5000
N52Z0.1000
(Contour-3)
N53G1Z-0.5000F4.0000
N54Y3.0072F8.0000
N55G2X1.4794Y2.5080I-0.5000
N56G1X1.4095Y2.5039
N57G3X0.2734Y1.5716I0.0734J-1.2478
N58G1X0.0227Y0.6107
N59X-0.4313Y0.7404
N60G0Z0.5000
N61M5
N62G0Z0H0
N63X0Y0E0
N64M2
(end)
```

REFERENCES

[ARSNRT98] Allen, R.H., et. al., Collaborating on the Design and Manufacturing of an Atomic Artifact Transport System: A Case Study in VRML as a Visualization Tool for Consensus Building, Proceedings of DETC'98 1998 ASME Design Engineering Technical Conferences, DETC98/DAC-5600, Sep. 1998.

[AZ89] Alting, L., and Zhang, H., Computer Aided process Planning: The State of the Art Survey, International Journal of Production Research, 1989, 27(4), 553-585.

[ALTINTAS94] Altintas, Y., and Munasinghe, W.K., A Hierarchical Open-Architecture CNC System for Machine Tools, Annals of the CIRP, 1994, 43, 349-354.

[CHANG90] CHANG, T.C., Expert Process Planning for Manufacturing, Addison-Wesley, ISBN 0-201-18297-1, 1990.

[CGT94] VERICUT 2.0 User Manual, CGTech, 1994

[CJ98] Choi, B.K., and Jerard, R.B., Sculptured Surface Machining - Theory and Applications, Kluwer Academic Publishers, ISBN 0-412-78020-8, 1998.

[CT90] Cutkosky, M.R., and Tenenbaum, J.M., Methodology and computational framework for concurrent product and process design, Mechanism & Machine Thery, 1990, 25(3), 365-381.

[CT92] Cutkosky, M.R., and Tenenbaum, J.M., Toward a framework for concurrent design., Journal of Systems Automation : Research an Applications, 1992, 1(3), 239-261.

[CTB92] Cutkosky, M.R., Tenebaum, J.M, and Brouwn, D.R., Working with Multiple Representations in a Concurrent Design System, Journal of Mechanical Design, 1992, 114(3), 515-524.

[CTG96] Cutkosky, M.R., Tenebaum, J.M., and Glicksman, J., Madefast: Collaborative Engineering Over the Internet, Communications of the ACM, Sep.1996, 39(9), 78-87.

[CPSC98] Chung, Y.C., Park, J.W., Shin, H. and Choi, B.K., Modeling the Surface Swept by a Generalized Cutter for NC Verification, Computer-Aided Design, Vol. 30, No. 8, pp 587-594.

[DGN95] Das, D., Gupta S.K., and Nau, D.S., Estimation of Setup Time for Machined Parts: Accounting for Work-Holding Constraints Using a Vise, ASME Database Symposium Computer in Engineering Proceedings of the 1995 Database Symposium, 85PIAS, Sep. 17-20,1995, 619-631.

[ECKS99] Eckstein, R., XML Pocket Reference, O'Reilly, 1999.

[MEAD81] Electronics, For Optimal VLSI design efforts, Mead and Conway have fused device fabrication and system-level architecture, Oct.1981.

[FS20] FeatureCam Software, Engineering Geometry Solutions, http://www.featurecam.com.

[FR2000]	Forrester Research, eMarketplaces Will Lead US Business eCommerce To $2.7 Trillion In 2004, According To Forrester, Feb. 2000, http://www.forrester.com/ER/Press/Release/0,1769,243,FF.html
[FJ98]	Fussell, B.K. and Jerard, R.B., Toolpath Optimization by Real-time Application of an Integrated Geometric/Mechanistic Model, Proceedings of the 1999 NSF Design and Manufacturing System Conference, Long Beach, California, Jan. 5-8.
[FJH00]	Fussell, B.K., and Jerard, R.B., and Hemmett, J.G., CNC Feed Velocity Selection for Sculptured Surface Machining, Proceedings of the 2000 NSF Design and Manufacturing System Conference, Vancouver, B.C. Canada, Jan. 3-6.
[FJH99]	Fussell, B.K., and Jerard, R.B., and Hemmett, J.G., Automatic 5-axis CNC feed-rate selection via discrete mechanistic geometric and machine model integration, Proceedings of the IFIP TC5 WG5.3 Conference on Sculptured Surface Machining, 1999
[FJHE99]	Fussell, B.K., and Jerard, R.B., Hemmett, J.G., and Ercan, M.T., Toolpath Feedrate Optimization: A Case Study, Proceedings of the 2000 NSF Design and Manufacturing System Conference, Vancouver, B.C., Canada, Jan. 3-6,2000.
[GAIN99]	Gaines, D.M., A Tool-Centric Approach to Designing Composable Feature Recognizers, Proceedings of ACM Solid Modeling Symposium, Ann Arbor, Jun. 1999, 97-107.
[GAHA99]	Gaines, D.M. and Hayes, C.C., A custom-Cut: A customizable feature recognizer, Computer-Aided Design, Mar. 1992, 31(2), 85-100.
[GINDY89]	Gindy, N.N.Z, A hierarchical Structure for form features, International Journal of Production Research, 1989, 27(12), 2089-2103.
[GDRN97]	Gupta, S.K., Das, D., Regli, W.C., and Nau, D., Automated Manufacturability Analysis: A Survey, Research in Engineering Design, 1997, 9(3), 168-190.
[GRN95]	Gupta, S.K., Regli, W.C., and Nau, D., Manufacturing Feature Instances: Which Ones to Recognize?, In ACM Solid Modeling Conference, 1995.
[HS98]	Hardwick, M., and Spooner, D.L., STEP Services for Sharing Product Models in a Virtual Enterprise, Proceedings of the 1998 ASME Design Engineering Technical Conference, Atlanta., DETC98/CIE-5518, Sep.13-16.
[HELD91]	Held, M., On the Computational Geometry of Pocket Machining, Springer-Verlag Berlin Heidelberg, ISBN 3-540-54103-9, 1991.
[HLA94]	Held, M., Lukács, G., and Andor, L., Pocket Machining Based on Contour-Parallel Tool Paths Generated by Means of Proximity Maps, Computer-Aided Design, 1994, 26(3):, 189-203.
[HOWE97]	Howe, J., SilverSmith: The SilverScreen C Programming Manual, Schoff Development Corporation, 1997.
[HGS98]	Huegel, J., Ganter, M.A., and Soorti, D.W., Telepresence for Distributed Rapid Prototyping: A VRML Approach, Proceedings of DETC'98 1998 ASME Design Engineering Technical Conferences, DETC98/CIE-5525, Sep. 1998.

[HW90] Hummel, K.E., and Wolf, M.L., Integrating expert systems with solid modeling through inter-process communications and the applications interface specification, Computers in Engineering 1990 Proceedings of the 1990 ASME International Computers in Engineering Conference and Exposition, Aug. 5-8, 1990, 355-360.

[JERARD98] Jerard, R.B., A Clean Interface for Design and Fabrication of Discrete Mechanical Parts Based on Manufacturing Features, Proceedings of the 1998 NSF Design & Manufacturing Grantees Conference, Long Beach, CA., Jan. 5-8, 1999.

[JECO98] Jerard, R.B., and Cox, S.J., FACILE:A Clean Interface For Design and Fabrication, Proceedings of DETC'98 1998 ASME Design Engineering Technical Conferences, Sep. 1998.

[JERY2000] Jerard, R.B., Ryou, O.H., Internet Based Fabrication of Discrete Mechanical Parts, Proceedings of the 2000 NSF Design & Manufacturing Research Conference, Jan. 3-6, 2000, Vancouver, British Columbia, Canada.

[JHDS89] Jerard, R.B., Hussaini, S.Z., Drysdale, R.L. and Schaudt, B. Approximate Methods for Simulation and Verification of Numerically Controlled Machining Programs, The Visual Computer, vol. 5, no. 6, 1989.

[KN92] Karinthi, R.R., and Nau, D., An Algebraic Approach to Feature Interactions, IEEE Transactions on Pattern Analysis and machine intelligence, APR.1992, 14(4).

[KKKKO98] Kim, C.Y., Kim, S.H., Kang, S.H., Kim, N., and O'Grady, P., Internet-Based Concurrent engineering: An Interactive 3D System with Markup, Proceedings of the 1998 ASME Design Engineering Technical Conference, Atlanta., DETC98/CIE-5522, Sep.13-16.

[KRAMER92] Kramer, T.R., A library of material removal shapes element volumes (MRSEVs), Technical Report NSTIR 4809, NIST, Gaithersburg, MD, Mar. 1992.

[KJ86] Kramer, T.R., and Jun, J., Software for Automated Machining Workstation, Proceedings of the 1986 International Machine Tool Technical Conference, Chicago, Sep. 1986.

[LAMA93] Laakko,T. and Mantyla, M., Feature modeling by incremental feature recognition, Computer Aided Design, 1993, 25(8), 479-492.

[MDSI] Manufacturing Data Systems, Inc. (MDSI), http://mdsi2.com/.

[MC20] MasterCAM, CNC Software, Inc, http://www.mastercam.com.

[MEAD80] Mead, C. and Conway, L., Introduction to VLSI Systems, Addison-Wesley, 1980.

[MFGQ00] MfgQuote.com, (www.mfgQuote.com).

[MORR00] Morrison, M., et al., XML Unleashed, Sams Publishing, 2000.

[MOSIS] MOSIS Service, Information Sciences Institute, University of Southern California, http://www.mosis.org/

[MS01] On-line manual-Visual C++ Programmer's Guide, Microsoft, http://www.microsoft.com.

[NRC99] National Research Council, Funding a Revolution: Government Support for Computing Research, NATIONAL ACADEMY PRESS Washington, D.C., 1999.

[PARA] Parasolids, Unigraphics Solutions Inc, http://www.ugsolutions.com/products/parasolid.

[RGN94] Regli, B., Gupta, S.K., and Nau. D.S., Feature Recognition for Manufacturability Analysis, In ASME Computers in Engineering Conference, ASME, Sep. 1994, 93-104.

[RGN95] Regli, B., Gupta, S.K., and Nau. D.S., Extracting Alternative Machining Features: An Algorithmic Approach, Research in Engineering Design, 1995, 7(3), 173-192.

[RUBIN 87] Rubin, S.M., Computer AIDS for VLSI Design, Addison Wesley Longman, ISBN: 0201058243, Jun. 1987.

[RYJE2001] Ryou, O.H, and Jerard, R.B., NCML: An Internet Compatible Data Exchange Format for Custom Machined Parts, Proceedings of the 2001 NSF Design & Manufacturing Research Conference, Jan. 7-10, 2001, Tampa, Florida.

[SSSMW96] Sarma, S., Schofield, S., Stori, J., MacFarlane, J. and Wright, P., Rapid Product Realization from Detail Design, Computer Aided Design, 1996, 28(5), 383-392,

[SDC] Schroff Development Corporation, http://www.schroff.com.

[SW96] Scofield, S., and Wright, P.K., IMADE: A Hierarchy for Intelligent Machines, Concurrent Product and Process Engineering, ASME MED, 1996, 1, 53-65.

[SHMA95] Shah, J.J., and Mäntylä, M., Parametric and feature-based CAD/CAM : concepts, techniques, and applications, John Willy & Sons, Inc, ISBN 0-471-00214-3, 1995.

[SHMA95-1] Shah, J.J., and Mäntylä, M., Parametric and feature-based CAD/CAM: concepts, techniques, and applications: Chap4. Feature Creation Techniques, John Willy & Sons, Inc, ISBN 0-471-00214-3, 1995, 105-144.

[SHMA95-2] Shah, J.J., and Mäntylä, M., Parametric and feature-based CAD/CAM : concepts, techniques, and applications:Chap11. Feature-Based Process Planning, John Willy & Sons, Inc, ISBN 0-471-00214-3, 1995, 187-218.

[SMN94] Shah, J.J., Mäntylä, M. and Nau, D., Advances in Feature based Manufacturing, Elsevier, 1994, 161-184.

[SMS99] Smith, J.P., McMains, S.A., and Sequin, C.H., SIF: A Solid Interchange Format for Web-Based Prototyping, Proceedings of the 1999 NSF Design and Manufacturing System Conference, Long Beach, California, Jan. 5-8.1999.

[SNC_P01] ISO/DIS 14649-1, Industrial automation systems and integration –Physical device control-Data model for computerized numerical controllers- Part 1: Overview and fundamental principles, 2000.

[SNC_P10] ISO/DIS 14649-10, Industrial automation systems and integration –Physical device control-Data model for computerized numerical controllers- Part 10: General process data, 2000.

[SNC_P11] ISO/DIS 14649-11, Industrial automation systems and integration –Physical device control-Data model for computerized numerical controllers- Part 11: Process data for milling, 2000.

[SNC_P111] ISO/DIS 14649-111, Industrial automation systems and integration –Physical device control-Data model for computerized numerical controllers- Part 1111: Tools for milling, 2000.

[SPTE] Spatial Technology, http://www.spatial.com.

[STEP_W] The Final Draft of ISO/FDIS 10303-224 Mechanical Product Definition for Process Planning Using Machining Features, 1998.

[UML00] OMG Unified Modeling Language Specification V1.3, Object Management Group, Inc. (http://www.omg.org), Mar. 2000

[VRML] VRML, ISO/IEC 14772-1, http://www.vrml.org.

[WW98] Wang, F.F., and Wright, P.K., Web-Based CAD Tools for a Networked Manufacturing Service, Proceedings of the 1998 ASME Design Engineering Technical Conference, DETC98/CIE-5517, Atlanta, Sep. 13-16.

[W3C1] World Wide Web Consortium (W3C), http://www.w3.org.

[W3C2] World Wide Web Consortium (W3C), XML Linking Language (XLink) Version 1.0- W3C Candidate Recommendation 3 July 2000, http://www.w3.org/TR/2000/CR-xlink-20000703/