

The University of South Australia

SCHOOL OF ENGINEERING

The Development of a Collaborative CAD/CAM System
(Volume 1)

by

Yung-Chou Kao

B. Sc. M. Eng. National Cheng Kung University, Taiwan
M. Sc. M. Eng. National Cheng Kung University, Taiwan

A dissertation submitted to fulfil
the requirements for the degree of Doctor of Philosophy

October 1997

To My Family

Table of Contents

Table of Contents i

List of Figures v

List of Tables vii

Abbreviations and Acronyms viii

Summary x

Signed Declaration xiii

Acknowledgments xiv

Chapter 1 Introduction 1

1.1 Background Information 1

1.2 Literature Review 2

1.3 Research Objectives 9

1.4 Research Methodologies 9

1.5 Significance of this Research 10

1.6 Organisation of the Thesis 11

Chapter 2 CAD Collaboration and Exploration of Remote Machining 14

2.1 Introductory Comments 14

2.1.1 Overview 14

2.1.2 Assistance by Collaborative Tools 14

2.1.3 CAD/CAM Application Enhancement 16

2.2 Feasible Configurations for CAD Collaboration 16

2.2.1 Overview 16

2.2.2 Collaborative Engine Block Design 19

2.2.3 Assistance by Multimedia, X-window Sharing Tool, and ISDN 24

2.2.4 Concluding Remarks of CAD Collaboration 28

2.3 Exploration of a Remote Machining Environment 30

2.4 Concluding Remarks 34

Chapter 3 Development of a Remote Machining Cell (RMC) 37

3.1 Introduction *37*

3.2 The Remote Machining Cell *37*

3.2.1 Overview *37*

3.2.2 Data Communication *37*

3.2.3 Tool Path and NC Program Generation *39*

3.2.4 Workpiece Loading and Unloading *39*

3.2.5 Unattended NC Milling Machining *40*

3.3 System Configuration *40*

3.3.1 Overview *40*

3.3.2 Hardware *40*

3.3.3 Software *41*

3.3.4 The Configuration *41*

3.4 Description of the RMC System Communication *43*

3.4.1 Overview *43*

3.4.2 The Description of DNC2 ..... *43*

3.4.3 The Description of XFS-32 ..... *47*

3.4.4 The Description of Network File System (NFS) *48*

3.5 Integration of the RMC System *49*

3.6 Concluding Remarks *52*

Chapter 4 *COCADCAM* -- a Collaborative CAD/CAM System 54

4.1 Introductory Comments *54*

4.2 The COCADCAM System *55*

4.2.1 Overview *55*

4.2.2 The System Configuration of COCADCAM *56*

4.2.3 The COCADCAM Program Structure *61*

4.3 The COCAD Subsystem *65*

4.3.1 Overview *65*

4.3.2 The Structure of the COCAD Subsystem Configuration and Communication *66*

4.3.3 The Relation between COCAD and a CAD/CAM System *69*

4.3.4 Data Communication Algorithm *71*

4.3.5 Data Interchange Format for COCAD Communication *76*

4.3.6 Realisation of COCAD *78*

4.4 The COMAN Subsystem *80*

4.4.1 Overview *80*

4.4.2 Toolpath Generation *81*

4.4.3 Postprocessing of the NC Program *84*

4.4.4 Linking a Remote Machining Cell *85*

4.5 Concluding Remarks *87*

Chapter 5 Illustrative Examples 88

5.1 Introductory Comments *88*

5.2 Application in Reverse Engineering *88*

5.2.1 Overview *88*

5.2.2 Reverse Engineering a Turbine Blade Collaboratively through LAN *89*

5.2.3 Reverse Engineering a Horn Pad through the Internet and ISDN *93*

5.3 Application in Metal Extrusion Die Surface CAD/CAE/CAM *98*

5.3.1 Overview *98*

5.3.2 Traditional Integrated Procedures *99*

5.3.3 Distributed Environment *101*

5.3.4 Collaborative Design of Forward Extrusion Die Surface *102*

5.3.4.1 Overview 102

5.3.4.2 Inlet and Outlet Profile Selection 102

5.3.4.3 Die Surface Optimisation 103

5.3.4.4 Toolpath and NC Program Generation 104

5.3.4.5 Remote Machining 106

5.4 Application on the Generation of Preliminary Assembly Sequence *108*

5.4.1 Overview *108*

5.4.2 Example of the Preliminary Assembly Sequence *108*

5.4.3 Collaborative Preliminary Assembly Sequence Design *110*

5.5 Concluding Remarks *116*

Chapter 6 Discussion **118**

Chapter 7 Conclusion and Future Direction **123**

Bibliography **127**

Publications List **139**

Appendices

Apendix A. Email Messages **A.1**

Apendix B. Source codes for the Remote Machining Cell **B.1**

Apendix C. Main Source Programs for *COCADCAM* Communication **C.1**

C.1 Source Codes of Local Client camrl *C.1*

C.2 Source Codes of Local Server camrs *C.3*

C.3 Source Codes of Global Geometry Server ggserv *C.7*

C.4 Source Codes of Global Server gserv *C.9*

C.5 Source Codes of Header Files *C.11*

C.6 Source Codes of Other Header Files *C.12*

C.7 Other Source Codes *C.13*

C.8 Source Codes of local client for ULTRIX 4.5 ..... *C.73*

Apendix D. Source Codes of *COCADCAM*: *COCAD* and *COMAN* **D.1**

D.1 Source codes of COCADCAM using GRAPL-IV in Anvil-5000 ..... *D.1*

D.2 Sample Source codes of COCADCAM using GRIP in UGII *D.160*

List of Figures

Figure 1.1 A typical CAD/CAM system 3
Figure 1.2 The growth of networked hosts 4
Figure 1.3 The growth of Internet 5
Figure 1.4 The common practice of CAD/CAM collaborative tools 8
Figure 2.1 The network system configuration 17
Figure 2.2 Geographical location between UniSA and NCKU 17
Figure 2.3 Configuration for data sharing in CAD collaboration 18
Figure 2.4 A typical engine block with four components 19
Figure 2.5 Collaborative design procedures of engine block 20
Figure 2.6 A snapshot showing the Hot-Link function 21
Figure 2.7 System configuration for CAD collaboration 25
Figure 2.8 The screen layout example 25
Figure 2.9 The collaborative procedures 27
Figure 2.10 Tool path of the connecting rod 31
Figure 2.11 A picture of the machined connecting rod 32
Figure 2.12 Flowchart of the control program for remote machining 33
Figure 2.13 The layout of a remote machining unit 33
Figure 2.14 Enabling tools for collaborative CAD/CAM and remote machining 36
Figure 3.1 The configuration of a remote machining cell 38
Figure 3.2 The system configuration of the remote machining cell 42
Figure 3.3 Three-layer structure of DNC2 43
Figure 3.4 The selected DNC2 subroutines 46
Figure 3.5 RS-232 connection between CNC and PC for DNC2 46
Figure 3.6 Relationship between RMC and COCADCAM 49
Figure 3.7 Flowchart of the RMC control and monitor process 51
Figure 4.1 System structure of COCADCAM 56
Figure 4.2 System configuration of COCADCAM 56
Figure 4.3 Socket connection flow for connection-oriented client-server model 58
Figure 4.4 A sample connection from client to server 59
Figure 4.5 A concurrent server passes connection to a forked child process 59
Figure 4.6 The concurrent server passes connection to another forked child process 60
Figure 4.7 Layout of the physical devices in COCADCAM 60
Figure 4.8 Program structure (functions 1-5) 63
Figure 4.9 Program structure (functions 6-9) 64
Figure 4.10 Program structure (sub-functions) 64
Figure 4.11 The structure of COCAD 66
Figure 4.12 The outline of UNIX XCALL system communication 68
Figure 4.13 Movement of data between GRAPL-IV and external program (EP) 68
Figure 4.14 Example of time lines for read-write data between GRAPL-IV and EP 69
Figure 4.15 The direct CAD/CAM interface process 70
Figure 4.16 CAD/CAM interface through neutral files or memory buffer 70
Figure 4.17 The role of COCAD data format among CAD/CAMs 71
Figure 4.18(a) Communication between camrl and camrs on creator's side 73
Figure 4.18(b) Communication between camrs and ggserv on creator's side 74
Figure 4.19(a) Communication between camrl and camrs on observer's side 75
Figure 4.19(b) Communication between ggserv and camrs on observer's side 76
Figure 4.20 The graphical environment of COCAD 79
Figure 4.21 Collaborated results from COCAD 80
Figure 4.22 Components of COMAN 80
Figure 4.23 The toolpath of a single free form surface without boundary curves 82
Figure 4.24 The toolpath of a single free form surface with boundary 83
Figure 4.25 The toolpath of a text engraved on a single free form surface 83
Figure 4.26 The command file menu linking remote machining facility 86
Figure 4.27 The command file contents linking remote machining facility 86

Figure 5.1 The turbine blade geometry 89
Figure 5.2 Screen layout of the creator 90
Figure 5.3 Co-edited result from the collaborative CAD session on a turbine blade 92
Figure 5.4 The toolpath of the collaboratively modified turbine blade surface 93
Figure 5.5 The image of a horn pad 93
Figure 5.6 The layout of the linked system (revised from Figure 4.7) 94
Figure 5.7(a) Creator's screen for horn pad coediting session 95
Figure 5.7(b) Observer's screen for horn pad coediting session (after coediting) 95
Figure 5.8 Extrusion processes 98
Figure 5.9 A sample streamlined extrusion die profile (from Kao and Lin 1995a) 99
Figure 5.10 Traditional modules 100
Figure 5.11 Procedures for integrated extrusion die surface CAD/CAM 101
Figure 5.12(a) Profiles selection and die surface optimisation (on screen of Designer A) 103
Figure 5.12(b) Optimised Die surface received from Designer A (on screen of Designer B) 104
Figure 5.12(c) Toolpath generation of the die surface (on screen of Designer B) 105
Figure 5.12(d) Toolpath received from Designer B (on screen of Designer A) 106
Figure 5.12(a)~(d) Graphical environment for the metal extrusion die surface CAD/CAM 106
Figure 5.13 The machined electrode 107
Figure 5.14 Pneumatic piston assembly (from Lin 1996) 109
Figure 5.15 A sample graph of preliminary assembly sequence (base part 6) 109
Figure 5.16 A graph of an unfeasible preliminary assembly sequence (base part 4) 110
Figure 5.17 An alternator assembly (from Lin and Hsu 1996) 112
Figure 5.18 The initial precedence matrix (from Lin and Hsu 1996) 112
Figure 5.19 The simplified precedence matrix (base part 13 or ⑨) 113
Figure 5.20(a) The creator's screen layout in collaborative design 114
Figure 5.20(b) The observer's screen layout in collaborative design 114
Figure 5.21 Graph of preliminary assembly sequence (abstracted from Figure 5.20(a)) 115
Figure 5.22 Graph of preliminary assembly sequence (abstracted from Figure 5.20(b)) 115

List of Tables

Table 2.1 Selective collaborative tools *15*
Table 2.2 Three possible networking routes from UniSA to NCKU *23*
Table 2.3 The Sample configuration for ISDN on camr1 ..... *26*
Table 2.4 Application summary *29*
Table 2.5 The sample data file *31*
Table 3.1 DNC2 subroutines *44*
Table 3.2 The parameter settings in CNC controller for DNC2 operation *46*
Table 4.1 The communication data format definition *77*
Table 4.2 The sample communication data *78*
Table 4.3 The CAM communication data examples *82*
Table 4.4 A sample keystroke file *84*
Table 4.5 The command file linking a remote machining cell *85*
Table 5.1 Abstracted messages for remote machining *107*
Table 5.2 The assembly precedence matrix, base part is 6 (from Lin and Hsu 1996) *109*
Table 5.3 Data example for collaborative preliminary assembly sequence design *111*
Table 5.4 Data transaction example for Figure 5.19(b) *113*

Abbreviations and Acronyms

ACL	Agent Communication Language
API	Application Programming Interface
APT	Automatically Programmed Tools
CAD	Computer Aided Design
CAD/CAM	Computer Aided Design and Manufacturing
CADD	Computer Aided Design and Drafting
CAE	Computer Aided Engineering
CE	Concurrent Engineering
CIE	Computer Integrated Enterprise
CIM	Computer Integrated Manufacturing
CNC	Computer Numerical Control
CSCW	Computer Supported Cooperated (Collaborative) Work
DNC	Direct Numerical Control
DNC2	Direct communication interface between a Fanuc controller and a PC
DWF	Drawing Web Format proposed by Autodesk
DXF	Data eXchange Format
GRIP	Graphics Interactive Programming supported by UGII
HTTP	Hyper Text Transfer Protocol
IGES	Initial Graphics Exchange Specification
NC	Numerical Control
STEP	Standard for the Exchange of Product model data - ISO 10303
UGII	Unigraphics CAD/CAM system
VRML	Virtual Reality Modelling Language
WWW	World Wide Web
CGI	Common Gateway Interface
ASM	Advanced Solid Modeller - a CAD software
camrl	local client program in COCADCAM
camrs	local server program in COCADCAM
COCAD	A Collaborative CAD system in COCADCAM
COCADCAM	A Collaborative CAD and Machining system in COCADCAM
COMAN	A Collaborative MAchiNing system
ggserv	global geometry server in COCADCAM

gserv	global server in COCADCAM
LASM	a copy of ASM software running on a local computer
RMC	Remote Machining Cell
SASM	A copy of ASM shared by cooperated engineers through 'shared application'
ARPANET	Advanced Research Projects Agency Network
FTP	File Transfer Protocol
Internet	The world-wide network
IP	Internet Protocol
IPC	InterProcess Communication
ISDN	Integrated Services Digital Network
ISO	International Organization for Standardization
LAN	Local Area Network
NFS	Network File System
NSFNET	the National Science Foundation's Network (USA)
OSI	Open Systems Interconnection
PC	IBM compatible Personal Computer
PPP	Point to Point Protocol
TCP	Transmission Control Protocol
CAMR	Centre for Advanced Manufacturing Research at UniSA
NCKU	National Cheng Kung University
UniSA	University of South Australia

Summary

The development of a collaborative CAD/CAM system is presented in this thesis. The system consists of two modules, namely, a collaborative CAD/CAM (*COCADCAM*) and a remote machining cell (*RMC*). *COCADCAM* has extended a single-location CAD/CAM to a multi-location application and *RMC* can be used to accept a numerical control (NC) milling task from the Internet and perform the machining operation at a distance. Two geographically dispersed CAD/CAM users can co-edit the three-dimensional CAD collaboratively and interactively through the *COCADCAM* system. *COCADCAM* has been implemented to use both the Internet and the Integrated Services Digital Network (ISDN). An intercontinental link has also been tested between Australia and Taiwan where the communication route went through more than twenty computer gateways across the United States and Japan.

The design philosophy behind the existing application of CAD/CAM is still based on its original computer environment, developed in the 1970s, which supported individual work, and was therefore restricted to a single-location application. Within these constraints, users cannot interact and collaborate on editing CAD models through their CAD/CAM system. In this research, it was determined that traditional serial development of a product, utilising CAD/CAM technology, may encounter some difficulties, such as a lengthy development time for design and manufacture, when compared with a collaborative approach. Even so, CAD/CAM technology has been successfully applied, over the past three decades, in the manufacturing industry, resulting in significant improvements in productivity and competitiveness. Furthermore, the trend towards global manufacturing had been resulted from the extensive international operation and the growth of world trade. Global manufacturing has spread as manufacturing companies, in one industry after another, started to operate on an international scale, for example, parts are manufactured and assembled in a number of countries. This has been even more important because of a sharp increase in multi-national and multi-location enterprises. In the top fourteen industrialised countries, their growth has more than trebled, from 7000 in 1969 to 24000 in 1995. The situation clearly requires the application of CAD/CAM technology through networked computers connecting people in an information-rich environment; thus, the need has arisen for a collaborative CAD/CAM system to assist international manufacturing activities.

Advances in processor and computing technology have led to the common practice of CAD/CAM collaboration being assisted by certain collaborative tools to establish a computer-supported cooperative working environment. Examples of these tools are:

- text-based messages such as text e-mail, text talk, and facsimile;
- audio-based messages such as voice mail, telephone and Internet phone;
- two-dimensional drawings such as blueprint drawing by facsimile and mail (by express delivery);
- asynchronous computer support such as multimedia e-mail, file transfer (FTP) and world-wide web (WWW) services; and
- synchronous computer support including shared applications, multimedia audio, white-board and video, Internet (video)phone, 3D geometry viewers, and desktop conferencing.

However, none of these focuses on the international linkage (for collaborative CAD/CAM) and the extension of a single-location CAD/CAM software to a multi-location application.

The underlying algorithm adopted in the *COCADCAM* system includes network technology and CAD/CAM functions. The network technology was derived from the implementation of UNIX inter-process communication, use of the Network File System (NFS) to access remote files, and a connection-oriented client and server model on top of the Transmission Control Protocol/Internet Protocol (TCP/IP). The CAD/CAM functions consisted of the generation of a milling toolpath, the output of a NC program following collaborative CAD discussion, and the facilitation of an environment for collaborative design, analysis, and manufacturing. Communication channels between the dispersed collaborators were assisted by selected collaborative tools. The *COCADCAM* system can be operated between remote locations underpinned by the application programming interface (API) of a single-location CAD/CAM with the implementation of the networking algorithm developed in this research.

This research has constructed a feasible framework to enable two specialist engineers to work together as a team on a three-dimensional computer-aided design so that each partner can provide synchronous contributions. This possibility has resulted in an application environment for international collaboration beyond physical barriers; a product can now be designed, tested and analysed on one side of the world, then manufactured on

the other side through the application of *COCADCAM*. The *COCADCAM* system has been successfully applied in

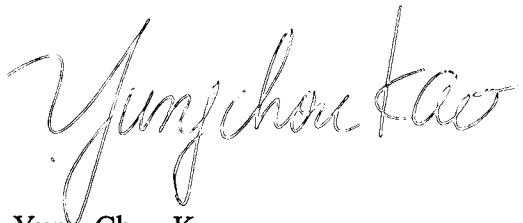
1) collaborative co-editing and revision of CAD geometry on the measurement data of an automotive component by a coordinate measurement machine for reverse engineering,
2) collaborative design on the preliminary assembly sequence, based on the choice of a different base part, and
3) a three-axis machining operation for the agreed geometry, linked to an active remote machining cell.

Therefore, the developed *COCADCAM* system can be applied to reduce the lead time toward global manufacturing.

Potential benefits for this system are the reduced lead time to market, the production of the right product in the right place, elimination of shipment time, increased competitiveness and profitability. Multinational companies can locate available worldwide expertise needed by a project at an international level to enhance the performance of the project.

Signed Declaration

I declare that this thesis does not incorporate without acknowledgment of any material previously submitted for a degree or diploma in any university; and that to the best of knowledge it does not contain any materials previously published or written by another person except where due reference is made in the text.



Yung-Chou Kao

Acknowledgments

I am truly grateful to the people who have supported and contributed to the thesis preparation during my Ph.D. study in the University of South Australia, South Australia, Australia.

First of all, I appreciate Professor Grier C. I Lin, the principal supervisor of my Ph.D. study, for his consistent support, guidance, encouragement and patience throughout this research. His comments and experiences are invaluable on my research.

Secondly, I would like to thank the encouragement from the Chief Editor, Suren N. Dwivedi, of a new journal of International Society for Productivity Enhancements (ISPE) entitled "Journal of Emerging Mechanical Engineering Technology." He invited me through E-mail, as listed in Appendix A, to be a member of the editorial board for this new journal. This acknowledges the research I have conducted, as reported in this thesis.

I would like to express my sincere thanks to my wife, Fei-Ying Huang, for her sacrifices, understanding, patience, love, and constant encouragement. I also thank my daughters, Yu-Hui and Yu-Chen, who had many days without proper care of their father. I would like also thank my parents who gave my family and me the love and support, which we needed most of all.

Thanks also go to the Study Advisers at the Levels campus, University of South Australia, Australia, for their suggestions in enhancing my thesis writing.

Finally I would like to thank my colleagues from the Centre for Advanced Manufacturing Research (CAMR), especially Hung-Yao Hsu for their valuable comments and discussions.

Chapter 1 Introduction

1.1 Background Information

Advances in computer and communications technologies have made computer assisted workgroups potentially the most convenient and powerful mechanism for geographically distributed groups of people wishing to communicate (Hiltz 1978, Press 1993). However, the design philosophy of existing CAD/CAM applications is still based on a computer environment, developed in the 1970s (Zientara 1981), which offered a tool that assisted in resolving the increased complexity in the engineering of new products and production facilities (Hofman 1993). Within this constraint, CAD/CAM software is a single-location application that is different from "groupware" (Ellis 1988, 1991, Marca 1992, Feik 1993a, Grudin 1994, Rogers 1994). Groupware was defined by Ellis (1991) as 'computer-based systems that support groups of people engaged in a common task (or goal) and that provide an interface to a shared environment' as opposed to most software systems that only support the interactive operation between a user and the user's computer system.

In the past three decades, CAD/CAM technology had been successfully diffused into manufacturing industries and resulted in significant improvements in productivity and competitiveness. Concurrent engineering (CE) had also been recognised as an enabling manufacturing philosophy (Kovács 1992, Dowlatshahi 1994) in the realisation that computer integrated manufacture (CIM) could not reach its full potential without CAD/CAM. CE emphasised a cooperative methodology between the engineering and manufacturing activities by offering real opportunities to create a synergy between them. Successful stories about concurrent engineering usually emphasised the links between many divisions in a company for sharing and exchanging ideas and experiences to reduce development cycle time and cost by resolving product, process and organisational issues at the early stage of design (Chen and Hsiao 1997). The NEON car development project from Chrysler Motors (Haase 1994) was a good example, which was finished in 31 months at a record cost reduction. Lutz (1994) made an advertising comment that Chrysler had speeded its new car to market in a new record time with a working style that employed a multi-functional team, cross-functional team, virtual team, or multidisciplinary task force (Hartley 1992). A critical characteristic for the success of concurrent product development was to have team members of different expertise effectively working together on a

common task. This was in contrast to a traditional serial *over-the-fence* or *over-the-wall* working style; for example, a design engineer defined all the CAD geometry, passed it to a drafting engineer in the drawing division for the engineering drawing to be passed to the manufacturing department. This process often resulted in tedious revision and wasted a lot of time. With the entire design team organised as a task force, the design engineer, the drafting engineer and the manufacturing engineer could sit *side-by-side* and work together if they were located in the same department. This resulted in a parallel and continuous development process, rather than a sequential one in which designs were passed back and forth (Deitz 1995, Reich 1992). Increasingly, product development involves teams of engineers from multiple organisations working together over networks, such as the Internet, supported by computation and information services. Therefore, in companies having multi-location operations, a potential need has arisen for a collaborative CAD/CAM computer system to facilitate a close teamwork on CAD/CAM application between geographically dispersed engineers.

1.2 Literature Review

The development of CAD/CAM was based on CAD, which could be traced back to the Automatic Programmed Tools (APT) project (Rembold 1993) started in 1950s at the Massachusetts Institute of Technology (MIT). A typical APT program consisted of four classes (McMahon 1993):

(i) geometry statements,
(ii) motion statements,
(iii) post-processor statements, and
(iv) auxiliary statements.

APT was a fixed text format created to represent the workpiece geometry for numerically controlled (NC) machines in achieving high-precision operations. APT was not, by its nature, designed for interactive operations. The CAD/CAM technology was designed as an interactive application for a single-user and therefore a CAD/CAM user could only communicate with a computer's central processing unit (CPU). Commercial CAD/CAM packages became available in the early 1970s, nearly at the same time (1969) that the Department of Defence (DoD) in the United States launched its Advanced Research

Projects Agency network (ARPANET) (Davison 1988); but the typical configuration of a CAD/CAM system was still designed as a stand alone system, as illustrated in Figure 1.1. The Internet came into existence as a natural development arising from the ARPANET and the National Science Foundation's network (NSFNET) in the 1980s (Krol 1993). Nevertheless, CAD/CAM users could still not synchronously communicate and collaborate on editing CAD geometry through CAD/CAM computer software, even though the CPU was typically a time-shared system linking several terminals running the CAD/CAM software. Local area networks (LAN) became popular and CAD/CAM systems began to be implemented on a network server, allowing users access from any computer station networked to the server. A network consisted of a group of computers, called nodes or hosts, connected to establish a communication link for data exchange amongst one another. A server was a host on a network that provided resources, such as disk storage, to a client host. Every host could then communicate, share databases and share output devices (Kennedy 1986). Despite these facilities, interaction among CAD/CAM users was still much the same as before except that more advanced communication services were increasingly used to shorten the time-span for communication. Such services included e-mail, ftp, facsimile, video phone, Internet phone, World-Wide Web (Kang 1997, Kim 1997), and overnight express delivery, etc.

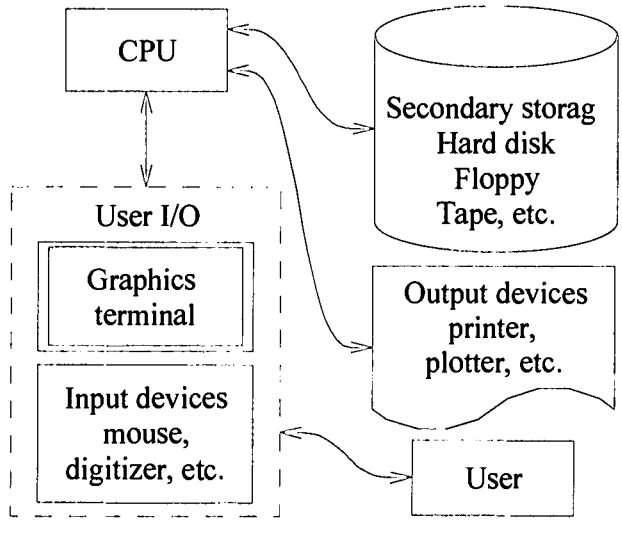


Figure 1.1 A typical CAD/CAM system

Global manufacturing had been very important as a result of the extensive international operation and the growth of world trade. The philosophy of global manufacturing (Miller 1992, Puri 1996) had spread as manufacturing companies, in one industry after another, started to operate on an international scale. In global manufacturing,

parts were manufactured and assembled in a number of countries. This has increasingly become more important because of the drastic increase of multi-national enterprises in multi-location operation. The manufacturing enterprises in the United States, for example, were forming strategic partnerships with diverse companies, geographically dispersed throughout the country, that were most effective at carrying out needed manufacturing operations (Bloom and Christopher 1997). In the top fourteen industrialised countries their growth had more than trebled (Lee 1995), from 7000 in 1969 to 24000 in 1995. Thus, it would be useful for CAD/CAM technology to facilitate concurrent engineering and assist international manufacturing activities for the computer integrated enterprise (CIE) in multi-location operations connecting CAD/CAM engineers in an information-rich environment. Moreover, the concepts of reach (world-wide connection without physical barrier) and compatibility (connection among heterogeneous computer platforms) were introduced by the Internet (Watson 1996), followed by the drastic increases in the number of networked computer nodes, as shown in Figure 1.2 (Hart 1995). There were already 9.5 million hosts on the Internet in January 1996, as shown in Figure 1.3 (Gray 1996), with more than three million hosts in the commercial company domain (Lottor 1996) and, at the connection level, most computers in the world have been Internet-enabled. This trend suggested a need for the application of CAD/CAM technology through networked computers, connecting people to other people in a geographically distributed environment for collaborative application. Thus, a collaborative CAD/CAM application was considered to be a worthwhile field for research.

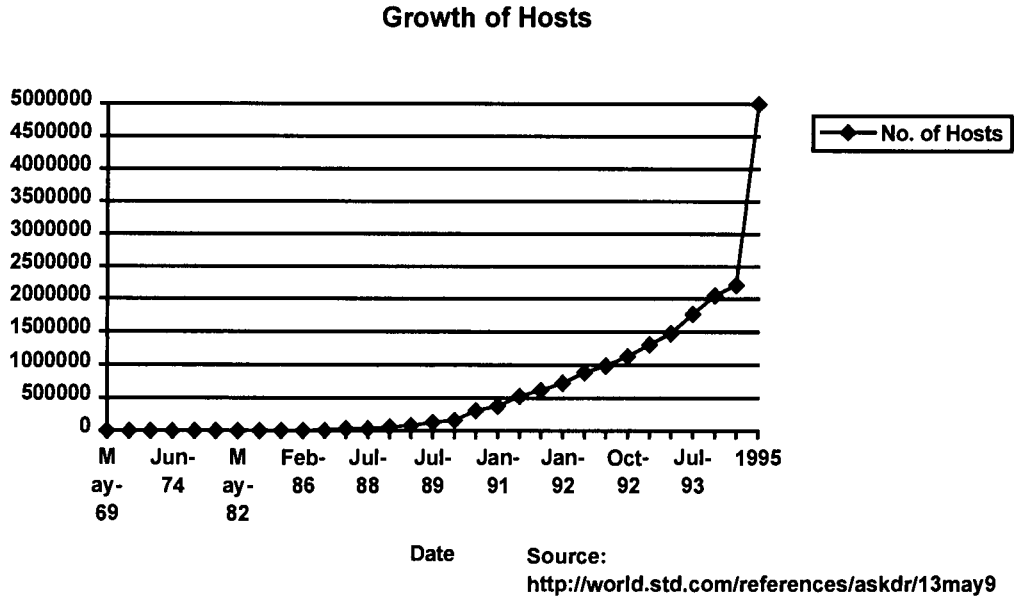


Figure 1.2 The growth of networked hosts

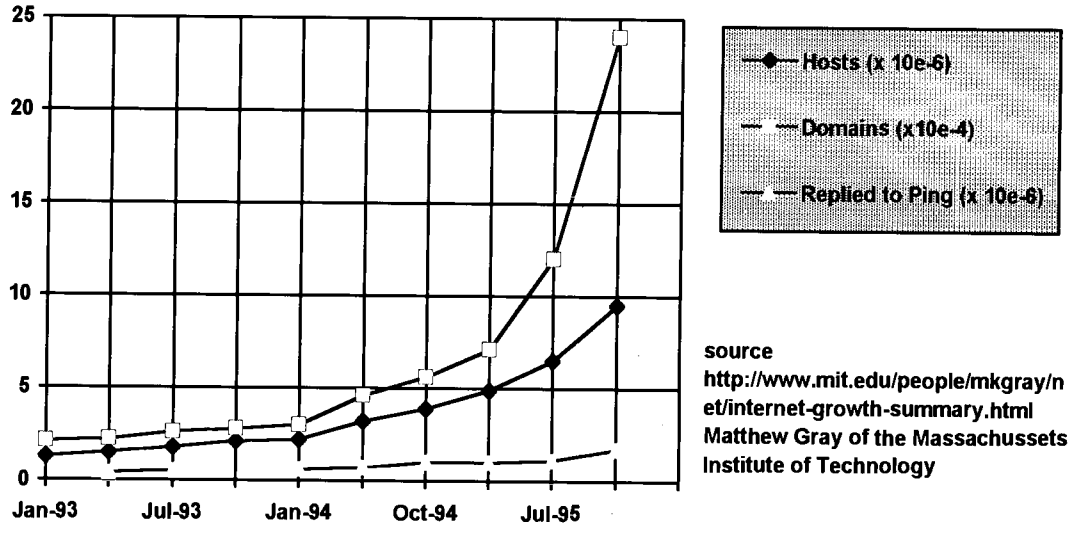


Figure 1.3 The growth of Internet

A collaborative CAD/CAM application should ideally provide designers, at two or more remote locations, with the capability of working together on a common design activity (Lin 1993). This implies that a new design environment is needed to make full use of all the relevant domain-expertise, technology and resources in the world. To achieve such a challenging task, current computer and network technology should be considered to enhance communication among designers. For example, multimedia, 2D and 3D whiteboard systems (SGI 1995, 1996, Kress 1996), world wide web browsers and viewers (Browsers 1996), and shared application tools, etc., have been shown to be very useful in enhancing the connection among geographically separated designers (Fraunhofer 1993, Geoview 1994, Jasnoch 1994, Toye 1994, Kress 1995, Lin 1995a, Sasse 1995, SGI 1995, Stratiquest 1996). However, the data transaction time through the Internet was very unreliable (Lin and Kao 1995a). For example, it took about five to sixteen minutes for 45155 bytes to go through the Internet but only a few seconds for 125589 bytes to go through a local area network in an experiment (Lin and Kao 1995a). The time periods in the experiment were measured when UNIX mount (Callaghan 1996) for file sharing through Network File System (NFS) was used. Lin and Kao (1995a) referred this to an inter-continental link between the Centre for Advanced Manufacturing Research (CAMR), University of South Australia, Australia, and the National Cheng Kung University (NCKU), Taiwan. This indicated that the data and information transmission through the Internet was not reliable. Thus express delivery, for example, Federal Express, was used and would still be used for a while (Salwen 1996a, 1996b).

A series of research efforts focused on Computer Supported Cooperative Work (CSCW) since 1980s (Sarin 1985, ACM 1986, 1988, 1990, 1992, Perreault 1991, Rodden 1991, Baecker 1993, Duffy 1995, Eltoweissy 1993, Feik 1993a, 1993b, Greif 1986, Malone and Crowston 1990, Michelis 1993, Connolly 1994, Rogers 1994, Spurr 1994, Udupa 1992). Most of these efforts concentrated primarily on creating an environment for co-authoring, documentation and message passing such as collaborative word-processing that mirrored the issues of collaborative CAD. However, only a few discussed CAD collaboration. For example, 'DistEdit' was developed to build interactive group editors for distributed environments (Knister 1990); 'VideoDraw' was a video-based prototype tool for sharing two dimensional sketch drawings (Tang 1991); 'TOPES' was developed for exchanging graphics, drawings and text (Pferd 1979); CAD conferencing was explored to exchange CAD drawings with video and whiteboard support (Newton 1993); 'ClearBoard' was designed to permit dispersed co-workers to draw with colour markers or with electronic pens and software tools while maintaining direct eye contact and the ability to employ natural gestures (Ishii 1993); 'Teledesign' was explored for examining three-dimensional CAD groupware interface issues (Shu 1992, 1994); 'CrystalPad' was developed for the creation and revision of documents (Peters 1995); 'Shared 3D viewer' was developed for the discussion on 3D CAD geometry file (Kress 1996); and 2D Drawing Web Format (DWF) was defined by Autodesk to collaborate, publish and manage design data over the World Wide Web (Autodesk 1996a, 1996b).

A collaborative environment was implemented by Abdel-Wahab et al (1988) providing a team that consisted of remote users on different computers with an underlying computer network to collaborate synchronously using familiar single-user tools. Abdel-Wahab et al (1991) also proposed a distributed prototypic framework called X-Terminal View (XTV), for synchronously sharing the user interface of X-window applications amongst a team of remotely-dispersed users at workstations running X-window and interconnected by the Internet. The XTV system could thereby enable single-user applications to be shared amongst networked users. Such research was categorised as a 'share application' and an event-driven application-sharing technology was also developed by Hao (1996) to reduce the network bandwidth caused by the 'share application'. A groupware tool-kit 'GROUPKIT' for developing a real-time groupware application was developed by Roseman (1993). This tool-kit tried to minimise the developer's work in constructing a groupware application. Similarly, JATLite (JAVA Agent Template, Lite)

was developed (JATLite 1997) in the platform-independent Java language that allowed users to quickly create new software agents that communicate robustly over the Internet.

The majority of the above research originated from the perspective of computer science. Although only drawings on two-dimensional graphics, three-dimensional solid primitives, or viewing (rotating, translating and scaling) and annotating (text, audio and video clips) on three-dimensional models (Bell 1995, ISO 1996, Schulman 1995, Young 1996, Kress 1996), which work asynchronously, were explored, these research efforts have reflected some potential problems in collaborative CAD/CAM. For example, most of the 'application sharing' resulted in heavy traffic load or demanded very high bandwidth in network data transmission. From the viewpoint of mechanical and manufacturing engineering, a still higher level system would be needed. Such a system should support the manipulation of CAD geometry that could be extended for collaborative and interactive co-editing (McCullough 1995) and CAM for manufacture, thereby forming a team working environment at a distance without consuming too much network bandwidth.

Current CAD/CAM systems primarily support interactive operations, between humans and computers, to increase the productivity of individual work lacking the connectivity between CAD/CAM users. Thus, the traditional serial development of a product (Erhorn 1994) is likely to encounter a few difficulties such as a lengthy development time for design and manufacture when compared against an integrated model such as a collaborative team approach. With the advancement of computing technology, the common practice for CAD/CAM collaboration was to establish a computer-supported cooperative working environment by certain collaborative tools (Schroeder 1993). Examples of these tools are listed as the followings:

- text-based messages such as text e-mail, text talk, and facsimile;

- audio-based messages such as voice mail, telephone and Internet phone;

- two-dimensional drawings such as blueprint drawing by facsimile and mail (includes express delivery);

- asynchronous computer support such as multimedia e-mail (Foo 1996), file transfer (FTP) and world-wide web (WWW) services; and

- synchronous computer support including shared application, multimedia audio, video and whiteboard, video phone through the Internet or telephone service provider, 3D geometry viewers, and desktop conferencing.

The application of such tools was categorised based on a four-square map of groupware options, as shown in Figure 1.4 (O'Hara-Devereaux 1994, Rogers 1994). The most effective collaborative work occurs in the same time and at the same place (face-to-face) using the physical presence of the participants to immediately clarify misunderstandings (Subrahmanian 1993, Levy 1993). In the absence of face-to-face contacts, individual engineers should be able to participate in an asynchronous manner (different-time versus different-place) through a variety of representational forms, media, and modes of communication, such as e-mail and facsimile, while retaining the time sequence of the exchanges. Different-time (asynchronous) and same-time (synchronous) collaboration were sometimes complementary and could be adopted selectively. However, none of these focused on both the international linkage for CAD/CAM collaborative practice together with the extension of a single-location CAD/CAM software to a multi-location application for geographically distributed CAD/CAM tasks.

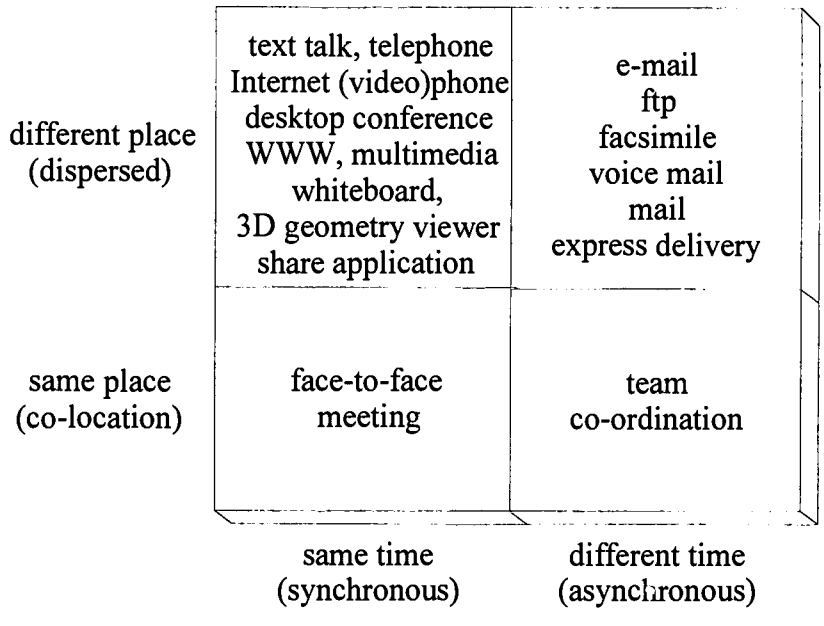


Figure 1.4 The common practice of CAD/CAM collaborative tools

1.3 Research Objectives

The objective of this research was to develop a distributed collaborative CAD/CAM system that could be operated remotely and linked through the Internet, or equivalent network, for international CAD/CAM collaboration. The collaborative CAD/CAM system (COCADCAM) should be able to provide a synchronous working environment for geographically dispersed CAD/CAM based tasks.

1.4 Research Methodologies

There were two steps determined in the development of the COCADCAM system based on the information from the literature review to achieve the research objectives. The first step was initiated to examine existing computer-supported collaborative systems, to derive the strategies for constructing the infrastructure of a collaborative CAD/CAM system among geographically dispersed designers. The application of computer-supported tools underlying the Internet and a local area network was then studied. Some new findings were expected to achieve the CAD/CAM collaboration by exploring the following issues.

(i) What are the networking requirements of collaborative tools?

(ii) What are the capabilities and limitations of current systems for computer-supported collaborative work?

(iii) What progress has been made by industry and academia in supporting the networking requirements of collaborative tools?

(iv) Which communication paradigms are applicable to collaborative processing, and what is their performance trade-off when implemented in current network and computer environment?

(v) How can current networking techniques be integrated, modified and enhanced in order to provide a solid, reliable communication system for collaborative applications?

(vi) What are the capabilities of current software in enabling a collaborative CAD/CAM?

These issues would not be answered directly but were considered as the basis towards the development of a collaborative CAD/CAM system in the second step.

The second step based on the findings from the first step to investigate the feasibility and to study the strategy in enabling organisationally and geographically dispersed CAD/CAM users an environment for CAD/CAM based collaborative work. For example, the editing of CAD geometry and related application collaboratively and interactively. A remote machining cell (*RMC*) was determined to be implemented to equip the CAD/CAM based collaborative work with physical remote machining facility. The CAD/CAM based collaborative work was determined to extend a traditional single-location CAD/CAM software to a multi-location application for geographically dispersed collaboration.

1.5 Significance of this Research

This research has initiated a new research field "collaborative CAD/CAM" in the world of CAD/CAM. The research area on the Computer-Supported Collaborative Work (CSCW) has been broadened by accommodating CAD/CAM in collaborative design, planning and manufacture. Remote manufacturing has also been explored in extending the collaborative CAD/CAM system (*COCADCAM*) developed in this research. The significance of this research are listed as the followings:

- The collaborative CAD/CAM system (*COCADCAM*), developed in this research, was the first system which injected a new paradigm of dynamic CAD geometry co-editing into the traditional single-location CAD/CAM. A system infrastructure was proposed and implemented to extend the traditional single-location CAD/CAM to multiple-location CAD/CAM application.

- A new CAD/CAM data transaction format was defined and implemented to facilitate the interactive and dynamic collaborative CAD/CAM work through *COCADCAM*. In general, neutral data exchange format, such as DXF, IGES and STEP, were designed for static data exchange rather than providing interactive and dynamic CAD data

revision. A neutral dynamic CAD/CAM data exchange format was expected for collaborative CAD/CAM work among heterogeneous CAD/CAM systems. This format must consider not only the CAD/CAM data but also the geographically distributed networking characteristics to ensure stable CAD/CAM product data exchange. Although the details of the new format is out of the scope of this research, it is envisaged to conduct further investigation.

- A dedicated communication medium, the Integrated Services Digital Network (ISDN), was implemented for reliable data transaction and showed promising results. Since the Internet had the nature of being publicly shared, there was no guarantee of the bandwidth sharing at any one time. The ISDN medium implemented in this research only had the transmission rate of narrow-band 64Kbit per second. Therefore, faster communication media, such as Frame Relay and Asynchronous Transfer Mode (ATM), were expected to shorten the time delay of geographically dispersed collaborative work.

- A remote machining cell (*RMC*), which could be accessed and monitored globally through the World Wide Web, was implemented for the three-axis milling machining at a distance. The integration of the *RMC* and the *COCADCAM* system could together reduce the manufacturing lead-time by making the component at the right place in the global manufacturing environment.

1.6 Organisation of the Thesis

There are two volumes in this thesis. Volume 1 consists of seven Chapters and published papers. This chapter presents an introduction to the background information supporting this research, such as the purpose of collaborative CAD/CAM. For example, the application of collaborative CAD/CAM can reduce the lead time towards global manufacturing when CAD/CAM technology is adopted. The network technology can be, and should be, applied to facilitate success for the global competitive business. The motivation of this research was supported by the structure of the traditional CAD/CAM application which, by its nature, did not support synchronous network communication channel amongst users. This chapter includes also the literature review, research objectives and methodologies.

There are two ways to enhance the traditional single-location CAD/CAM application for multi-location collaboration. One is the adoption of collaborative tools to wrap the CAD/CAM software into an integrated and distributed application. The other is to develop new modules and functions to enhance the single-location CAD/CAM software for multi-location application. Both methods are described in Chapter 2.

Chapter 2 presents an environment for CAD collaboration and explores the function of remote machining for collaborative CAM. The CAD collaboration was achieved with the assistance of collaborative tools and with the enhancement of the CAD software that was adopted. An engine block assembly, with four components, was adopted to demonstrate the environment of collaborative CAD, linking geographically dispersed designers in Australia and Taiwan, through the Internet. Multimedia and application sharing tools were explored to enhance the communication channel in the collaboration. The exploration of remote machining is not to focus on the technology but to study the feasibility in extending the remote machining capability to be operated through the Internet.

The research documented in Chapter 3 aims at developing a remote machining cell (*RMC*) by extending the findings documented in Chapter 2. The strategies to develop the *RMC* are presented. The configuration of the *RMC* is described, which consisted of one machining centre for machining operation and one robot performing the automatic loading and unloading tasks on the requested workpiece. However, there is a missing link, that is the CAM function, between the CAD collaboration presented in Chapter 2 and the physical machining presented in Chapter 3. This missing link is resolved and illustrated through the development of a collaborative CAM module in the *COCADCAM* (collaborative CAD/CAM) system which is presented in Chapter 4.

In Chapter 4 the *COCADCAM* system is proposed as a means for collaboratively editing CAD/CAM data including CAD geometry and CAM information. *COCADCAM* consisted of two subsystems: *COCAD* (CO-editing CAD) and *COMAN* (CO-operative MAchiNing). *COCAD* adopted a socket-based connection-oriented client and server model for reliable network communication, and allowed geographically dispersed users to view and to collaboratively co-edit CAD geometry. *COMAN* could generate milling toolpath and NC program of a free form surface for three-axis machining. The system configuration and the development of *COCADCAM* are illustrated and explained.

Chapter 5 documents the application in integrating the remote machining cell, as presented in Chapter 3, and *COCADCAM*, as developed in Chapter 4, to form an international collaborative CAD/CAM system for CAD co-editing, machining, and discussion and revision on a design. This chapter concentrates on showing the integrated configuration instead of focusing on the individual technology. Examples are used for the illustration, which include a reverse engineering application, a distributed metal forming extrusion die surface CAD/CAE/CAM, and the application of collaborative planning on preliminary assembly sequence design. The example on the reverse engineering shows the capability of the *COCADCAM* system on the final stage refinement in revising a CAD model collaboratively. The next example shows the extension of a traditional integrated metal forming system to a distributed and collaborative application: the design and analysis on one side of the world, and the machining process on the other side. The last example presents a feasible working environment of collaborative planning for the distributed discussion and design of the preliminary assembly sequence. These examples cover the application on the revision of CAD geometry (drawing), remote machining (manufacturing), and collaborative planning on the generation of preliminary assembly precedence (design).

Chapter 6 is the discussion session. The advantages, disadvantages, characteristics, potential applications and problems are discussed. These include the network response time, the bandwidth availability, the resource sharing underlying the Internet, and the growth of the World-Wide Web (WWW) services.

Chapter 7 presents the conclusions and the future directions. Several key issues are emphasised and proposed for the further enhancement of the implemented system. Some of the important issues are the integration to WWW browsers and viewers, data base management and the CAD data underlying the new networking environment, Hyper Text Transfer Protocol (HTTP) application, Virtual Reality Modelling Language (VRML) application, and integration with intelligent software agents. The bibliography then follows and the hard copy of publications generated as a result of this research are attached.

Volume 2 consists of the Appendices. Appendix A lists the E-mail messages that the author was invited to be a member of a new Journal during this research. The source codes that were generated as a result of this research are listed in Appendix B ~ D and are stored in a diskette attached at the back of this thesis.

Chapter 2 CAD Collaboration and Exploration of Remote Machining

2.1 Introductory Comments

2.1.1 Overview

The practice of manufacturing globalisation has led to a possible need of a collaborative CAD/CAM system in reducing the manufacturing lead-time. A computer-supported collaborative working (CSCW) environment is explored to support CAD/CAM collaboration in an international link from collaborative design to remote machining. This chapter presents the construction of such an environment to fulfill collaborative CAD/CAM tasks using available computer and communication technology. Collaborative tools, such as multimedia, desktop conferencing and telecommunications were adopted for the interpersonal communication. The application of UNIX X-window application sharing tools, to enhance the collaborative work, is also explained. A connection-oriented client-server program based on the Transmission Control Protocol/Internet Protocol (TCP/IP) was developed to start a real time interactive collaboration; and a special function was implemented to facilitate an existing CAD package for dynamic CAD collaboration. The remote machining function was explored to study the feasibility in extending the CAD collaboration to a CAM application. The feasible configurations were then suggested to enhance the existing CAD software for collaborative application.

Two methods are determined and adopted to establish a computer-supported cooperative environment for CAD/CAM collaboration based on available network and computing resources. One is the assistance by collaborative and groupware tools, and the other is the enhancement of CAD/CAM application software. These methods are described in Sections 2.1.2 ~ 2.1.3.

2.1.2 Assistance by Collaborative Tools

The assistance from collaborative or groupware tools can enhance the interpersonal communication, which can be any combination of the following technology: e-mail (Hypermail), multi-media, shared whiteboard (screen), network audio (conference

telephone), Internet (video)phone, video conferencing, shared application, World-Wide Web (WWW) hypertext browsers and viewers (navigator), 3D CAD geometry viewers, Mark-up and annotator (design review), etc (Greenberg 1991, Kress 1996). These tools can form a feasible environment of multimedia conferencing through the Internet and ISDN (Sasse 1994, Schulman 1995). For example, Designer A can develop a CAD model and e-mail it to Designer B after making (attaching) annotations such as text, audio and video files to a proper position in the CAD model. Designer B, then can amend the CAD model, by referencing the annotation files, and e-mail the model back to Designer A or to another Designer C. Anything that can be digitised, such as audio, full-motion or still-video, manuscripts, data, and charts, can be captured, combined, manipulated, compressed, stored, retrieved, clipped and used for the annotation. However, this is an asynchronous operation mode; the designers can not concurrently and interactively co-edit the CAD geometry even though they may use advanced e-mail systems with automatic answering capability. The reason is that, on the one hand, *the file size of a CAD model is usually large while e-mail was not designed for sending or manipulating big files*; on the other hand, e-mail may not be sent instantly and it may take several days to arrive at the destination. This happened, in the link, from the University of South Australia in Australia (UniSA) to the National Cheng Kung University (NCKU) in Taiwan (Lin and Kao 1995b), when the bandwidth was limited and the Internet load was very heavy. Furthermore, at the final stage of a design where there is only a little geometry to be changed, it is neither economical nor necessary to send the whole large CAD file again to the partner. Therefore, a possible application would be to combine some of the collaborative tools, as shown in Table 2.1, which is presented in Section 2.2 to assist a collaborative work.

Table 2.1 Selective collaborative tools

Function description	Tools
SharedApplication (share X-Window)	TeamSolutions <Platform Independent> InPerson <SGI> <Unix and Windows> XTV: synchronously share X-window application ShowMe SharedApp <Sun UNIX>
Audio communication	ShowMe Audio, InPerson Audio, etc.
Graphic communication	ShowMe whiteboard, InPerson whiteboard, etc.
Video communication	ShowMe Video, Inperson Video, etc.
CAD exchange and communication	InPerson 3D whiteboard, Mark-up, annotation TeamExchange, Shared 3D viewer, VRML viewer, etc.

2.1.3 CAD/CAM Application Enhancement

Enhancement of the CAD/CAM software is the other way to enable a feasible computer-supported cooperative environment to extend a single-location CAD/CAM to a multi-location application such as co-editing CAD geometry. The co-editing of CAD geometry is the focus of Chapter 4 on the development of a collaborative CAD/CAM system (*COCADCAM*). Software modules or third party software must be developed and applied to facilitate the existing CAD/CAM software for multi-location application.

2.2 Feasible Configurations for CAD Collaboration

2.2.1 Overview

To explore the feasibility of a computer-supported cooperative environment for international CAD/CAM collaboration, an experiment was launched and a system configuration was established, as depicted in Figure 2.1 (Lin, Kao and Lee 1994). This experiment used a client-server model to link the Centre for Advanced Manufacturing Research (CAMR), University of South Australia (UniSA) in Australia, and the Tjing-Ling Manufacturing Technology Development and Research Centre, National Cheng Kung University (NCKU) in Taiwan. The geographical location of this link is shown in Figure 2.2. Each site was considered as a separate subgroup and was composed of three workstations linked by the campus local area network (LAN). Both groups were also linked by the Internet, and the communication between these locations adopted text message passing and oral conversation through the conference telephone. The text *'talk'* function in Sun SPARC workstation was used as a tool for real time text conversation under the normal network load. Telephone was used for oral communication, as suggested by Resnick (1993). The CAD software "Advanced Solids Modeller" (ASM), developed by Qikdraw Systems Pty Ltd (Qikdraw 1993), was adopted because Qikdraw was a partner in the experiment.

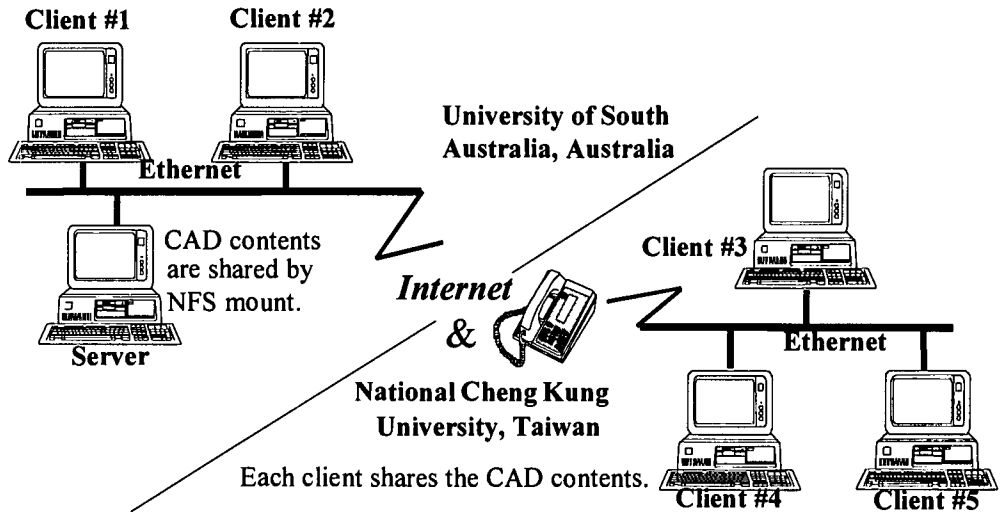


Figure 2.1 The network system configuration

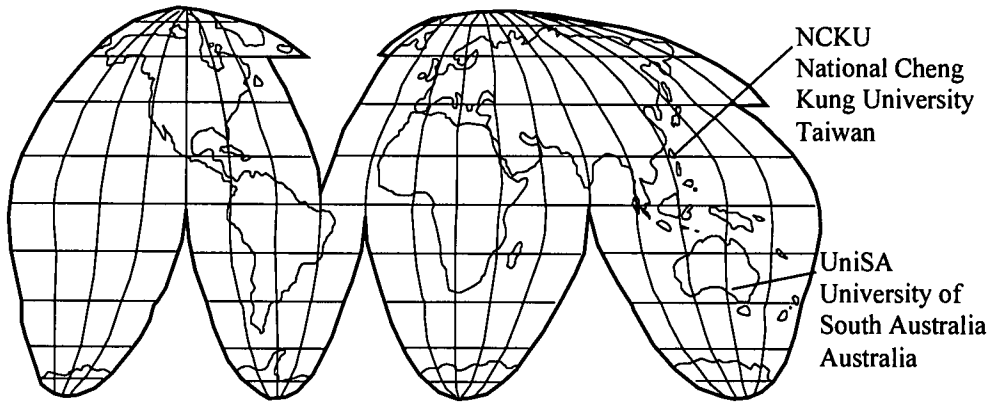


Figure 2.2 Geographical location between UniSA and NCKU

Two examples were used to study the feasibility of a collaborative CAD/CAM application based on a traditional computing environment, where computer, network and telecommunication technologies were adopted as the enabling tools. The data transmission environment, as shown in Figure 2.1, was based on the TCP/IP protocols (Santifaller 1991). This environment was used to support data communication (Davison 1988). The system configuration for the CAD data sharing, as shown in Figure 2.3, was fulfilled by mounting the common database through UNIX Network File System (NFS) which is described in Section 3.4.4. The directory DB1-1 was shared and used as the common database repository. Every designer, including the server, ran a local copy of ASM (LASM) separately. Subgroups 1 and 2 worked on their own design until they were notified to join the CAD collaboration work. The first example, as will be presented in Section 2.2.2, used traditional computing resources while the second example, as will be described in Section 2.2.3, took the advantage of collaborative tools.

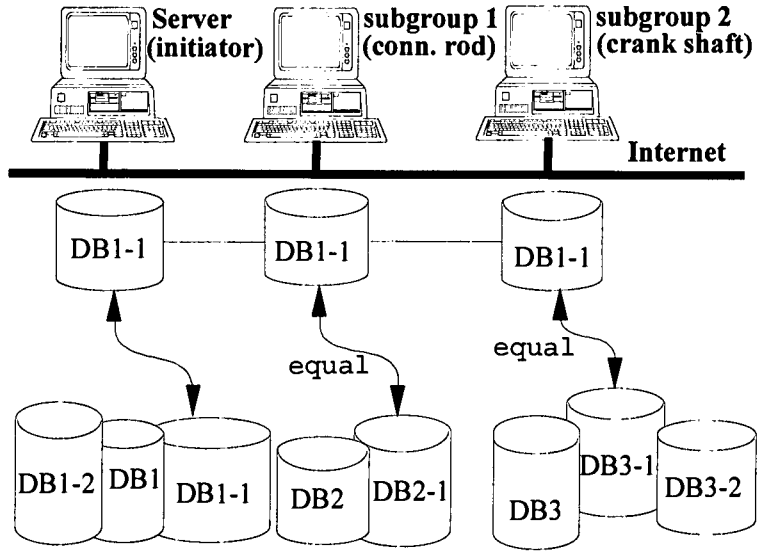


Figure 2.3 Configuration for data sharing in CAD collaboration

2.2.2 Collaborative Engine Block Design

To verify the feasibility of a computer-supported international CAD collaboration, a typical engine block with four components, as shown in Figure 2.4, that consists of *cylinder lining*, *piston*, *connecting rod*, and *crank shaft*, was adopted for an experimentation (Lin and Kao 1995b). These components were represented as CAD objects in the ASM software individually. The role of *Client #2*, as indicated in Figure 2.1, was assigned to assist *Client #1* in forming the subgroup 1 which was a subgroup on *connecting rod* design. Similarly, *Client #3*, *Client #4* and *Client #5* formed the other subgroup in designing the *crank shaft*.

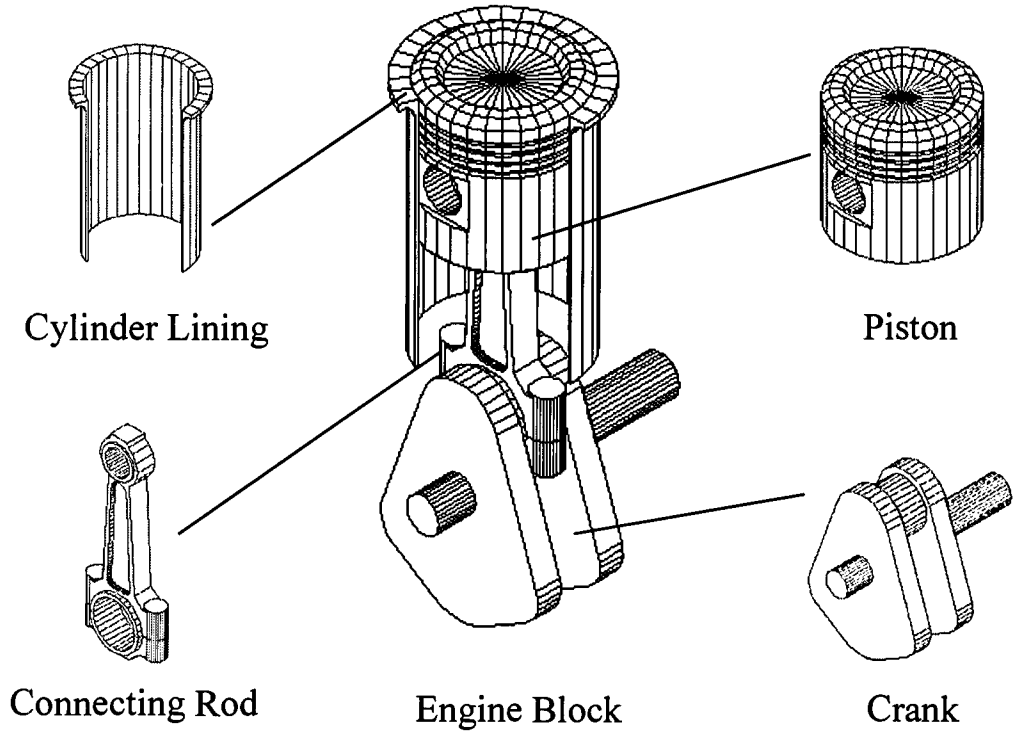


Figure 2.4 A typical engine block with four components

There were five steps in the CAD collaboration after every participant was ready. Firstly, the *server*, who was also the coordinator, rang *Client #3* to start the international collaborative work on engine block design. The *server* then showed a draft design of *cylinder-piston* assembly on his own screen (Step 1 in Figure 2.5) and asked for the contribution of *connecting rod* from *Client #1*. A text message was sent to *Client #1* from the *server* by using the 'talk' program in Sun workstation. Telephone (or direct oral conversation if the *server* and *Client #1* were located at the same place) was also adopted as a communication media. Responding to the need from the *server*, *Client #1* contributed

its design of *connecting rod* and sent a message back to the *server* immediately after the *connecting rod* was contributed to the shared common data base.

Secondly, the *server* accessed the *connecting rod* from the common database and attached it to the assembly, as indicated in Step 2 of Figure 2.5. Then the *server* asked *Client #3*, who was in Taiwan, over an international telephone conversation, to contribute the *crank shaft* component for the collaborated engine block. *Client #3* contributed the *crank shaft* into the common database and informed the *server* as soon as the contribution was finished. The *server* then attached the *crank shaft* component to assemble the engine block, as indicated in step 3 of Figure 2.5.

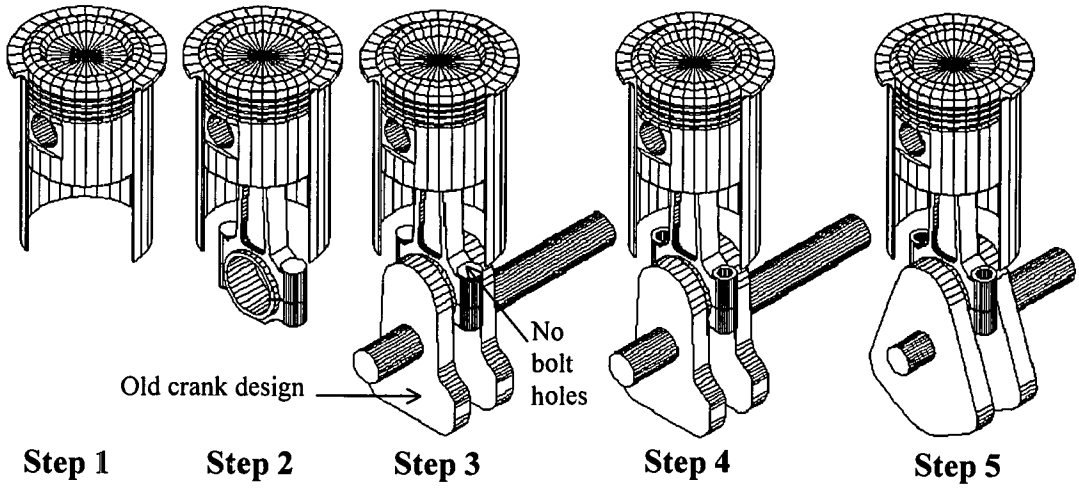


Figure 2.5 Collaborative design procedures of engine block

The draft design of the engine block assembly consisting of the *cylinder*, *piston*, *connecting rod* and the *crank shaft*, as indicated in Step 3 of Figure 2.5, could now be seen by all the collaborative designers that were geographically separated. However, *Client #1* did not consider the bolt holes that should be on the *connecting rod* and the *crank shaft* component contributed by Client #3 was considered an old design. Therefore, the *server* asked *Client #1* to modify the *connecting rod*, and requested *Client #3* to contribute new design alternatives for the *crank shaft*. This meant that the engine block design had come to the stage of an unavoidable dynamic collaboration between the *server* and *Client #1*, and between the *server* and *Client #3*, separately.

To achieve dynamic CAD collaboration, a request to monitor and update CAD object was suggested to the vendor company, after several trials of the international CAD collaborative work, and a special function called *Hot-Link* was added to the ASM software. The function *Hot-Link* was designed to maintain the consistency of the collaborative CAD data for the dynamic collaboration. The *Hot-Link* function was actuated to update the *Hot-Linked* CAD objects whenever the screen display was repainted. This function enabled the automatic revision of the designed object from the common database repository if the function '*Silent*', as shown in Figure 2.6, was selected.

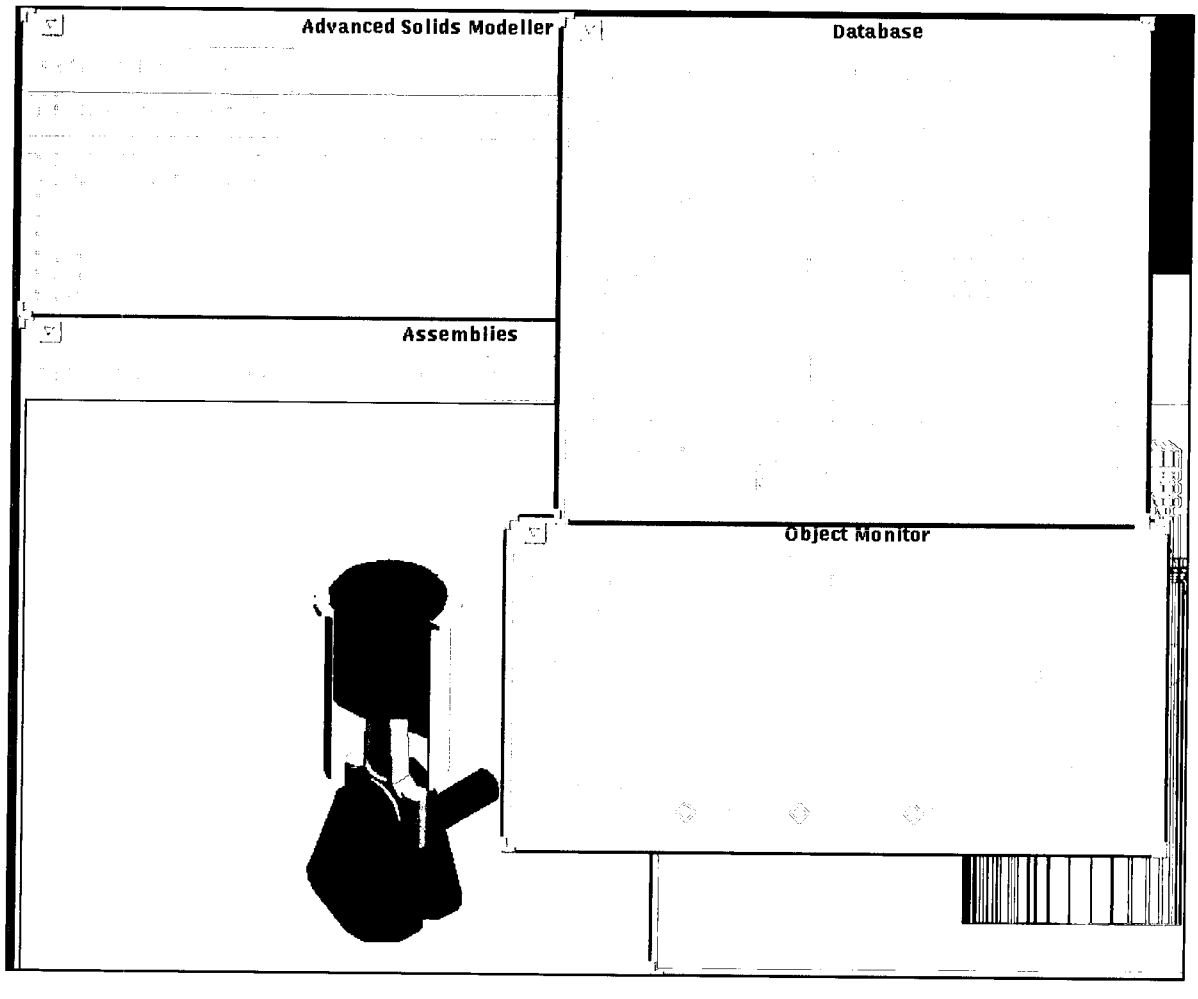


Figure 2.6 A snapshot showing the *Hot-Link* function

The server then set the *connecting rod* to be silently *Hot-Link*ed, as shown in Figure 2.6, and notified *Client #1* to modify the design of the *connecting rod. Client #1* revised the *connecting rod* immediately on his own local ASM software and contributed the modified *connecting rod* into the common database again. With the *Hot-Link* function, the *server* did not need to access and update the modified *connecting rod*; the ASM software

automatically updated the *connecting rod*, as indicated in Step 4 of Figure 2.5, where bolt holes were added.

Similarly, the *server Hot-Linked* the *crank shaft* before requested *Client #3* to contribute an alternative *crank shaft* object. The revision of the engine block was thus accomplished after *Client #3* contributed a new *crank shaft* into the common database. The on-line collaborative engine block design, the real-time *connecting rod* modification (under negotiated agreement), and design alternatives of *crank shaft* were demonstrated. This was an experiment to simulate a simplified CAD collaboration on the design activity of an engine block assembly.

The data transmission speed through the Internet fluctuated and was uncontrollable in the experiment, although the geographically dispersed designers were successfully linked. The data transmission speed depended heavily on the Internet traffic load shared by millions of users all over the world (Ishida 1993). Also, three possible Internet routes, as shown in Table 2.2, from UniSA to NCKU went through more than twenty Internet-networked computer gateways dispersed on three continents: Australia, America and Asia. Although the file sizes of the old and new *crank shafts* were 45155 bytes and 43121 bytes respectively, the approximate transmission time varied from five minutes to sixteen minutes. (The time was measured manually by the program /usr/openwin/bin/clock on Sun workstation.) On the other hand, the file size of the *connecting rod* was 125589 bytes which took about four to five seconds including software operation time for the data transmission in the local area network. Therefore, a more reliable transmission medium, for example, Integrated Services Digital Network (ISDN) was suggested. Furthermore, there were difficulties in the dynamic discussion when the *server* asked *Client #1* to make the bolt holes on the connecting rod because the communication channel only used audio medium. A more sophisticated (high performance) medium was apparently needed to allow the geographically dispersed designers to view the images, graphics, or CAD geometry in a faster and more effective way.

Having successfully experimented the international link of collaborative CAD, new technology of the collaborative tools, such as the maturing multimedia, were also explored. Through the collaborative tools, the CAD images could be discussed via the assistance of the shared whiteboard and audio link. A practical environment can be implemented, by applying these collaborative tools, in a LAN environment that does not have the unreliable

network bandwidth similar to the Internet. This is described in Section 2.2.3 which also uses ISDN as a dedicated transmission medium for the data transmission.

Table 2.2 Three possible networking routes from UniSA to NCKU

(Trumpet HOPCHECK - Copyright (c) 1992 by Tattam (1992))
(Traceroute (1995) 1.2 - Lawrence Berkeley National Laboratory, Network Research Group)

Node	Host name (in Dec. 1994) by Trumpet HOPCHECK (Tattam 1992)	Host name (in Aug. 1996) by Trumpet HOPCHECK (Tattam 1992)	Host name (in Aug. 1996) by TRACEROUTE 1.2 (Traceroute 95)
1	lvgtwy.levels.unisa.edu.au (130.220.17.1)	lvgtwy.levels.unisa.edu.au (130.220.17.1)	LVgate.Levels.UniSA.Edu.Au (130.220.17.1)
2	ungate.Underdale.UniSA.Edu.Au (130.220.36.6)	ungate.Underdale.UniSA.Edu.Au (130.220.36.6)	ungate.Underdale.UniSA.Edu.Au (130.220.36.6)
3	ntgate.City.UniSA.edu.au (136.169.26.2)	ntgate.City.UniSA.edu.au (136.169.26.2)	ntgate.City.UniSA.edu.au (136.169.26.2)
4	sa.gw.au	ethernet2-1.central.saard.net (136.169.248.1)	ethernet2-1.central.saard.net (136.169.248.1)
5	national-sa.gw.au (139.130.40.1)	fddi0-0.exchange.saard.net (203.21.37.2)	fddi0-0.exchange.saard.net (203.21.37.2)
6	usa.gw.au	serial3-2.way1.Adelaide.telstra.net (139.130.254.193)	serial3-2.way1.Adelaide.telstra.net (139.130.254.193)
7	border4-hssi1-0.SanFrancisco.mci.net	Fddi0-0.way-core1.Adelaide.telstra.net (139.130.237.226)	Fddi0-0.way-core1.Adelaide.telstra.net (139.130.237.226)
8	core-fddi-1.SanFrancisco.mci.net	Serial5-5.pad-core2.Sydney.telstra.net (139.130.239.21)	Serial5-0.lon-core1.Melbourne.telstra.net (139.130.239.21)
9	core-hssi-2.Denver.mci.net	Fddi0-0.pad8.Sydney.telstra.net (139.130.249.213)	Hssi6-0.pad-core1.Sydney.telstra.net (139.130.239.21)
10	core-hssi-3.Washington.mci.net	cpe4-hssi4-0.SanFrancisco.mci.net (204.70.204.17)	Fddi0-0.pad8.Sydney.telstra.net (139.130.249.213)
11	core-hssi-3.NewYork.mci.net	border2-hssi3-0.SanFrancisco.mci.net (204.70.33.9)	cpe4-hssi2-0.SanFrancisco.mci.net (204.70.204.9)
12	border2-fddi0-0.NewYork.mci.net	core1-fddi-1.SanFrancisco.mci.net (204.70.3.161)	border2-hssi3-0.SanFrancisco.mci.net (204.70.33.9)
13	jvncnet-ges-ds3.NewYork.mci.net	core3.SanFrancisco.mci.net (204.70.4.17)	core1-fddi-1.SanFrancisco.mci.net (204.70.3.161)
14	130.94.34.242	somwrouter.sprintlink.net (206.157.77.66)	core3.SanFrancisco.mci.net (204.70.4.17)
15	192.83.196.1	gsl-stock-3-Fddi0/0.gsl.net (204.59.128.197)	somwrouter.sprintlink.net (206.157.77.66)
16	192.83.166.1	204.59.160.54	gsl-stock-3-Fddi0/0.gsl.net (204.59.128.197)
17	134.208.200.112	192.83.166.1	204.59.160.54
18	134.208.206.110	MOE-T3-NCHC.edu.tw (134.208.214.110)	192.83.166.1
19	134.208.208.116	NCHC-T1-NCU.edu.tw (134.208.208.116)	MOE-T3-NCHC.edu.tw (134.208.214.110)
20	140.116.1.253	140.116.254.6	NCHC-T1-NCU.edu.tw (134.208.208.116)
21	140.116.254.6	140.116.175.2	140.116.254.6
22	140.116.175.2		140.116.175.2

2.2.3 Assistance by Multimedia, X-window Sharing Tool, and ISDN

To overcome the unreliable bandwidth of the Internet, LAN and ISDN were explored further for the CAD collaboration. ShowMe and SharedApp (ShowMe 1996) were added to enrich the collaborative environment. ShowMe was a multimedia tool that had three modules: Audio, Whiteboard and Video. The ShowMe Audio, for voice communication, instead of the ShowMe Video was adopted to save network bandwidth, as suggested by Chen (1992), because audio should be given higher priority over video. ShowMe Whiteboard was used as the graphics discussion panel, by snapping still-images, among the conferees. SharedApp was a tool to share single-user applications under X windows among the networked computers, e.g. COMIX (Babadi 1993) and XTV (Abdel-Wahab 1991). SharedApp was used for the conferees to watch the progresses of the CAD collaboration.

The system configuration, as shown in Figure 2.7, consists of three designers: *camr1*, *camr2* and *camr3*. Each designer accessed their own interactive workstation and ran Advanced Solid Modeller (ASM) separately as a daily design tool. In this configuration, SUN Sparc workstations were used as network nodes. The directory DB1-1, as shown in Figure 2.7, was shared by *camr1*, *camr2* and *camr3* as the common database repository. Every designer, including the *initiator* (chairperson), ran their local ASM, namely *LASM1*, *LASM2* and *LASM3* separately. *Camr2* and *camr3* worked on their own design until they were notified to join the session of the CAD collaboration. A set of connection-oriented server and client programs were developed for the notification to start the CAD collaboration. The appointment for the CAD collaboration was made by e-mail in advance, and the client program was used to notify the participants, running the server program, to start the multimedia tools for the collaboration. Later, the *initiator* invited *camr2* and *camr3* into the *ShowMe* and *SharedApp* conference. The "ShowMe Audio" software was used for oral communication and the "ShowMe Whiteboard" was used as the graphics discussion panel. The Whiteboard was used to snap and attach the image from the display screen onto the Whiteboard window itself. The participants used their mouse pointers to *mark* or *annotate* the region of the snapped image for visual discussions. The colours of the mouse pointers were arranged to be *red* for *camr1*, green for *camr2*, and blue for *camr3*, respectively, for easy recognition in the collaborative work, as shown in Figure 2.8.

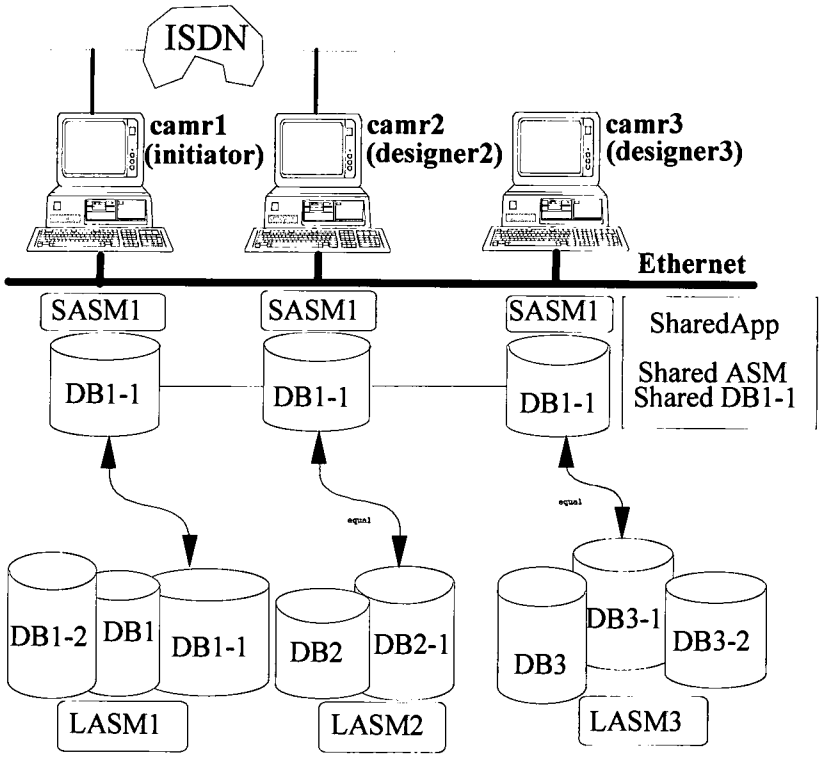


Figure 2.7 System configuration for CAD collaboration

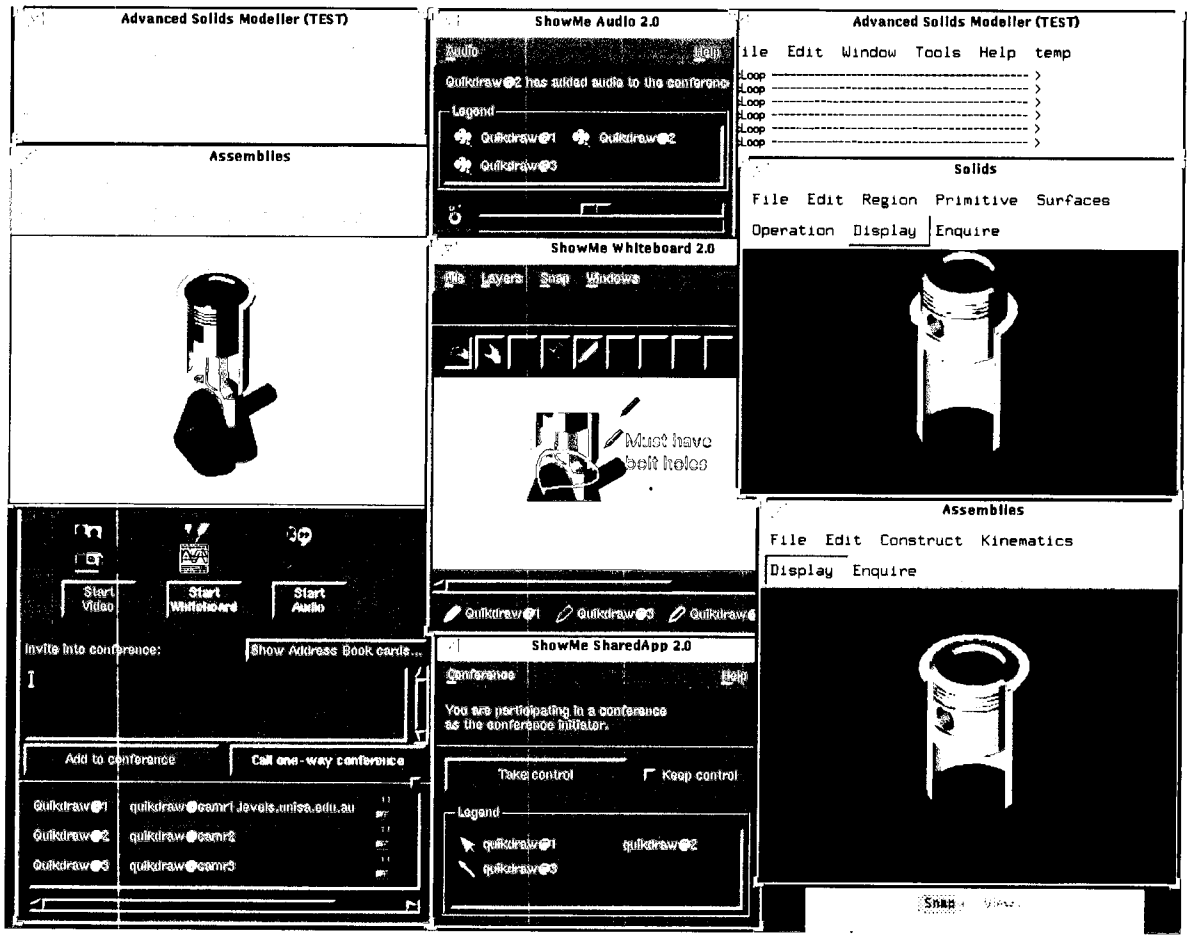


Figure 2.8 The screen layout example

Every node in this system, as shown in Figure 2.7, was linked through the campus Ethernet local area network; and a narrow-band ISDN dedicated subnetwork was also used to link *camr1* and *camr2*. The ISDN was a basic-rate (narrow band) on-board interface which was supplied with point-to-point protocol (PPP) software. The bandwidth was 10 Mbits per second over the campus local area network and 1B (64 Kbps) over the ISDN subnetwork of the telephone service provider Telstra Australia. The usage of ISDN was similar to that of the Internet which was transparent to the user. Therefore, it allowed user to run non-broadcast IP applications in the same way as that through the Internet or a local area network. This was supported through SunLink ISDN subsystem (SunLink 1993a, 1993b, Sun ISDN 1994). A sample ISDN configuration on the computer node *camr1* is listed, as shown in Table 2.3. The computer nodes *camr1* and *camr2* had the other host names *camr1i* (IP address 192.168.10.76) and *camr2i* (IP address 192.168.10.77) respectively which must be added into the file */etc/hosts*. The ISDN connection from *camr2i* to *camr1i* was similar to a dial-up connection from a home computer to its Internet service provider through normal telephone line.

Table 2.3 The Sample configuration for ISDN on camr1

Global properties: ISDN network type: Telecom Australia B-channel baud rate (optional) : 64K
Interface properties: Category: Basic properties for the ISDN Physical interface: isdn_0 ISDN phone number: 84065531 (ISDN subaddress:) Connection to network: Direct SunOS generic device name: /dev/isdn/0
IP/ISDN properties: Category: Global
Category: Remote IP Remote IP or path: 130.220.19.77 84065530 CONNECTION ESTABLISHMENT INFORMATION Remote ISDN address (ISDN phone number: 84065530, ISDN subaddress:) CONNECTION SETTINGS IP/dialup interface: ipd0 Physical interface: isdn_0 Protocol: IP/PPP Interactive timeout: 30 PPP MRU: 1500
IP/ISDN properties: Category: Routing command Routing command /usr/sbin/ifconfig ipd0 plumb 192.168.10.76 netmask 255.255.255.0 up

The collaboration was initiated by the *initiator*, that is to say, *camr1* was the chairperson in the CAD collaboration session. *SharedApp* was used for 3D CAD discussion in compensating the insufficiency of 2D Whiteboard panel on CAD discussion. To access the shared ASM (*SASM*), the designer must 'Take control' of the *SASM*; only the *initiator* can keep the control of *SASM*. A sample screen layout of the devised system environment is shown in Figure 2.8. Only part of the engine block image was snapped and attached onto the Whiteboard window for a graphics-rich discussion, as shown in the middle of Figure 2.8. The assembly window of *SASM* (at the middle-left screen of Figure 2.8) was available for further discussion through SharedApp and allowed for real time access by all the participants. The collaborative procedures are illustrated in Figure 2.9.

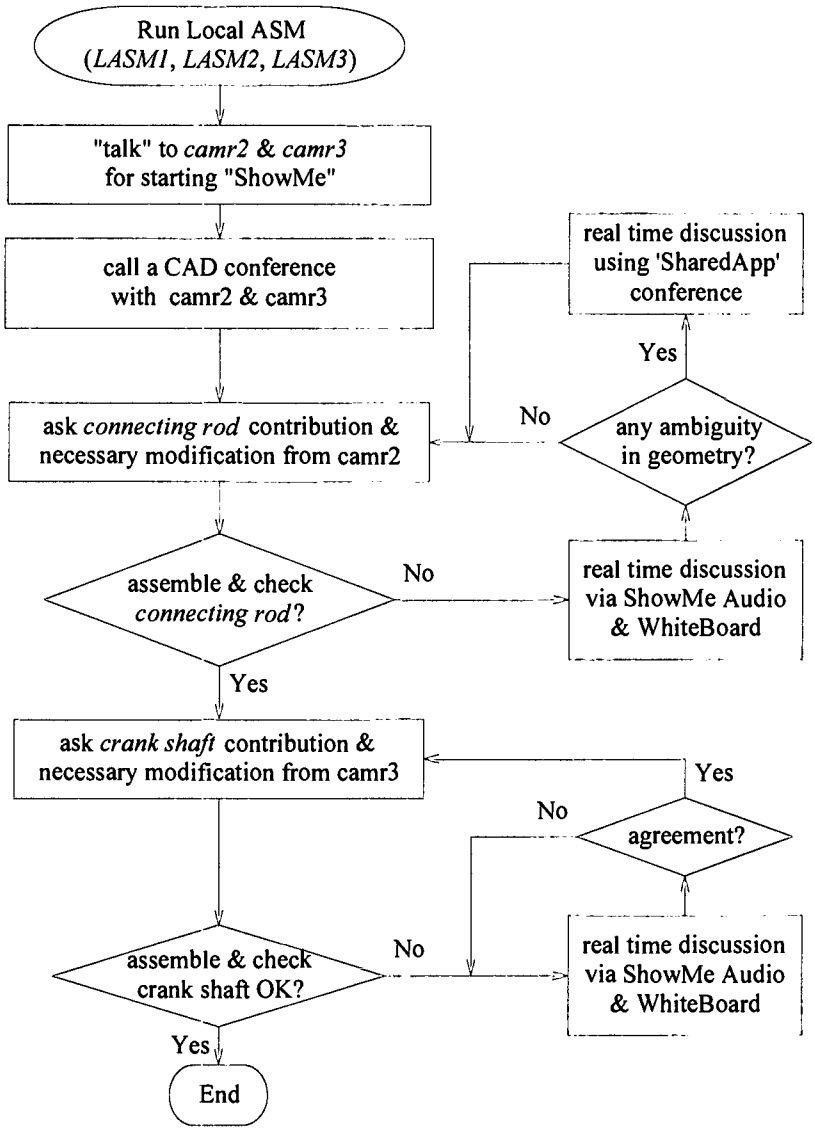


Figure 2.9 The collaborative procedures

The implementation of the ISDN subnetwork was very promising. The data transaction time through the ISDN subnetwork was very reliable compared with Internet; it took only a few seconds for the local ISDN data transmission rather than several minutes, as described in Section 2.2.2. Although the data transmission through ISDN was not as fast as that through a local area network, it was very reliable in the experiment. The usage of ISDN was transparent to the user and was the same as that of using the Internet in this implementation.

2.2.4 Concluding Remarks of CAD Collaboration

A feasible computer supported cooperated environment for CAD collaboration has been implemented using computing and networking resources. This has demonstrated that a collaborative CAD application linking geographically dispersed CAD/CAM users is achievable. The computing and networking environment is summarised in Table 2.4 and is explained as follows.

- The common database was mounted based on the UNIX network file system (NFS) to share the data transparently through a local area network, the Internet and the ISDN.

- A dedicated narrow-band ISDN (64 Kbits per second) was implemented and aimed to resolve the unreliable Internet bandwidth, which showed very reliable result for the network data transmission. A broadband ISDN can be implemented for heavy-loaded data transmission; and the networking speed will then be faster than the Internet that is shared by the public.

- The collaborative tools, for example, multimedia, X-windows sharing application, and network conversation tools have been applied and shown very useful and feasible applications to establish a collaborative CAD environment.

- A function that is called '*Hot-Link*' was suggested from the trials and was implemented by the ASM software vendor in dynamically updating the CAD data for the traditional single-location CAD software ASM to adapt the multi-location application environment.

- Multi-location organisations are encouraged to devote more consideration to the creation, management and use of collaborative tools, for example, interactive multimedia, rather than on the technology itself.

- Organisations should start to explore and learn more interactive communication skills using the opportunities provided by the existing collaborative tools. For example, online services delivered by telephony, interactive multimedia through ISDN, application sharing tools through the Internet or ISDN, and videoconference (Erhorn 1994), etc. The exploration of interactive communication skills will become valuable experiences of the organisation to enable easier migration to collaborative tools using wider bandwidth in the near future.

Table 2.4 Application summary

Data sharing	UNIX Network File System MOUNT
Networking (TCP/IP)	Local Area Network (LAN), Internet, and Integrated Services Digital Network (ISDN)
Conversation tools	text 'talk', conference telephone and *ShowMe* Audio
Discussion tools	*ShowMe* Audio and Whiteboard, and *SharedApp* to share X-window application
CAD software	Advanced Solid Modeller (ASM): 1) CSG solid modeller 2) special feature: *Hot-Link* for dynamically updating CAD data
Platform	Sun SPARCstation 10
Operating System	SunOS 4.1.3 and Solaris 2.4
ISDN software	*SunLink* ISDN 1.0 subsystem

2.3 *Exploration of a Remote Machining Environment*

Shortening production lead time is one of the major issues that must be resolved to improve the competitiveness of an enterprise. The production centre of a product, on one side of the world where the market place is situated, must sometimes accept the finalised design from the other side of the world where the major design is carried out. This provides an opportunity for lead-time improvement. However, such an infrastructure must be based upon the advanced computers and communications technology.

This section explores a remote machining environment (Kao and Lin 1995a) to study the feasibility in extending the international CAD collaborative work to have remote machining facility that is operable through network. This environment was achieved by networking a personal computer (PC), that was controlling a CNC machining centre, to the collaborative CAD server computer. The communication between the PC and the collaborative CAD server (Sun sparc workstation) was achieved through sharing the data file via XFS-32 (Juhasz 1994) from the collaborative CAD server. The description of XFS-32 is presented in Section 3.4.3. A server program was developed for the collaborative CAD server based on a connection-oriented client and server model for synchronous communication. This server program was designed to keep listening to an agreed well-known TCP port in order to respond to the requests from the client immediately for the deployment of a machining task. The PC controlled and monitored the machining centre directly through the DNC2 communication interface (Fanuc 1991a, 1991b) according to the command file from the collaborative CAD server. The DNC2 communication is discussed further in Section 3.4.2.

To focus on the exploration of a remote machining environment, the machining processes were assumed to have been determined. The electrode for electric discharging the mould of the *connecting rod* component which was discussed in Section 2.2.2 was chosen to show a 3D machining operation. The CAD geometry of the *connecting rod* was output from ASM in the Data eXchange File (DXF) format. The toolpath, as shown in Figure 2.10, was then created in ANVIL-5000 CAD/CAM package based on the DXF data from ASM. This toolpath was postprocessed to generate the numerical control (NC) program. The postprocessing parameters were customised to adapt the format of FANUC 0MF controller of the machining centre. Therefore, the NC program was ready to be downloaded into the machining centre for the NC machining operation.

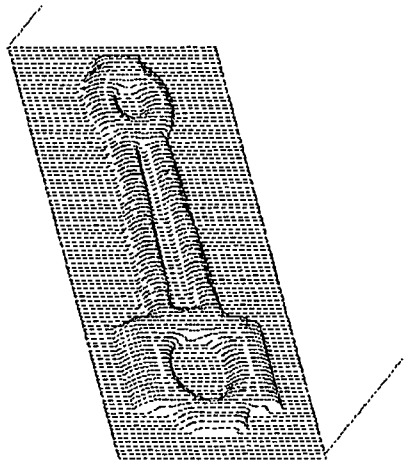


Figure 2.10 Tool path of the connecting rod

A data file was used for exchanging the information to control and to monitor the machining operation. This data file was created after the collaborative design was finalised. A sample data file is shown in Table 2.5 where the filename is MCVKCR and *'Machine'* represents the name of the CNC machine. *'Operation'* means an activity that the machine is being asked to perform. This operation can be *'CUT'* or *'CHECK'* the workpiece, *'OPEN'* or *'CLOSE'* the automatic door, *'CLAMP'* or *'UNCLAMP'* the air-operated vice, *'CLOSE'* or *'OPEN'* the air-actuated chuck, *'FORWARD'* or *'BACKWARD'* the air-operated tailstock, etc. *'Axis'* can be either three or four axes. Three-axis machining uses the air-operated vice as the clamping device. If *'Axis'* is '4' then four-axis machining would be actuated and the clamping device would be the automatic chuck mounted on the rotary table. *'NC_Filename'* is the NC program name to cut the component. *'Organisation'* means the organisation that issues this machining operation. *'Country'* means the location of the

Table 2.5 The sample data file

Filename	MCVKCR	
Contents	Example 1	Example 2
Machine	*MCV1000CR*	*MCV1000CR*
Operation	*CUT*	*CUT*
Axis	*3*	*4*
NC_Filename	*ROD*	*CAMSHAFT*
Organisation	*NCKU*	*CAMR*
Country	*Taiwan*	*Australia*
Time	*13:50*	*10:30*
Date	*25 Jan 1995*	*27 Jan 1995*

organisation. *'Time'* and *'Date'* are the time and date that this data file was lodged. The data file was stored in a common database repository. Example 1 shows a command file requesting for the machining task of the *connecting rod*, which was issued through the Internet, for example, from NCKU, Taiwan. A workstation located at the NCKU was remotely logged on and this command file was created and stored in the common database repository which was mounted through the network file system (NFS). The workpiece of the *connecting rod* was a rectangular block and was clamped by the VICE for three-axis machining. Example 2 shows a command file from the Centre for Advanced Manufacturing Research (CAMR) in Australia, through LAN for four-axis machining. A picture of the machined *connecting rod* is shown in Figure 2.11.

The data file MCVKCR, which was shown in the 'Example 1' of Table 2.5, could be created from the ASM software based on the ASM development kit and saved into the common database. A control and monitoring program was executed on a PC, which was a server of the machining centre, to check the data file that was located in the common database for the command message. The flowchart of the program is shown in Figure 2.12. The machining centre was actuated, in this example, by the *'CUT'* command. The three-axis machining operation was issued and the NC program file *'ROD'* was accessed. During the machining operation, the control program monitored the machining operation regularly by requesting the status of the machining centre through DNC2 interface communication. The machining operation continued in normal operation until either the machining task was finished or abnormal status occurred. The layout of the machining unit is shown in Figure 2.13.

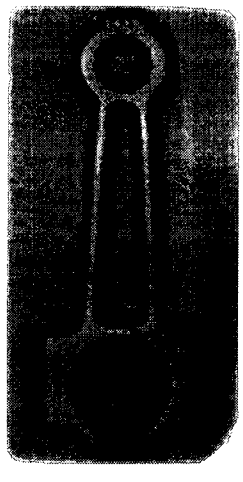


Figure 2.11 A picture of the machined connecting rod

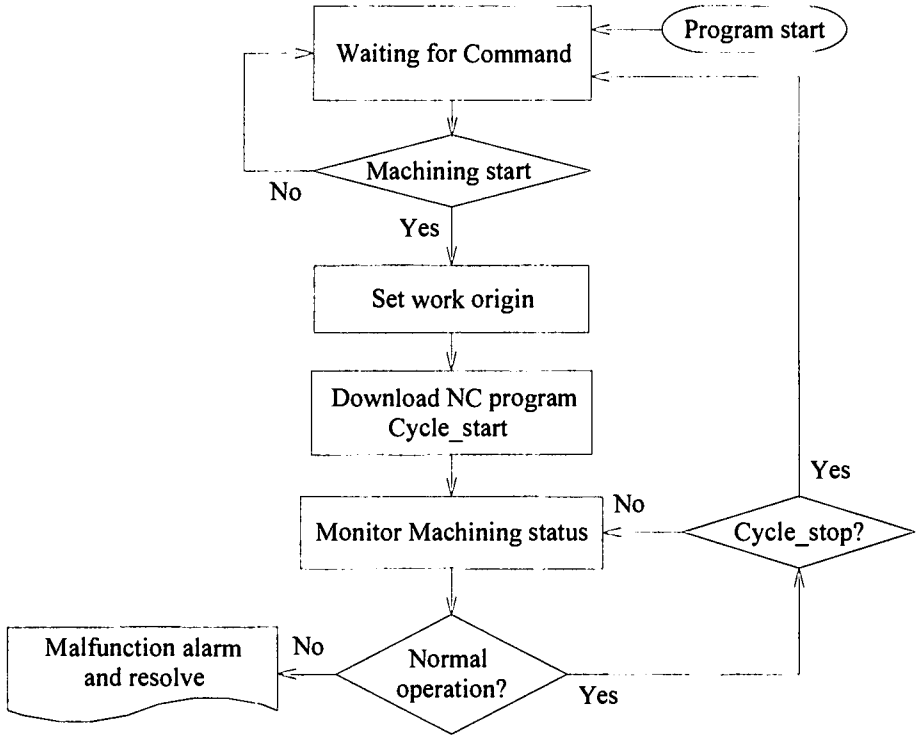


Figure 2.12 Flowchart of the control program for remote machining

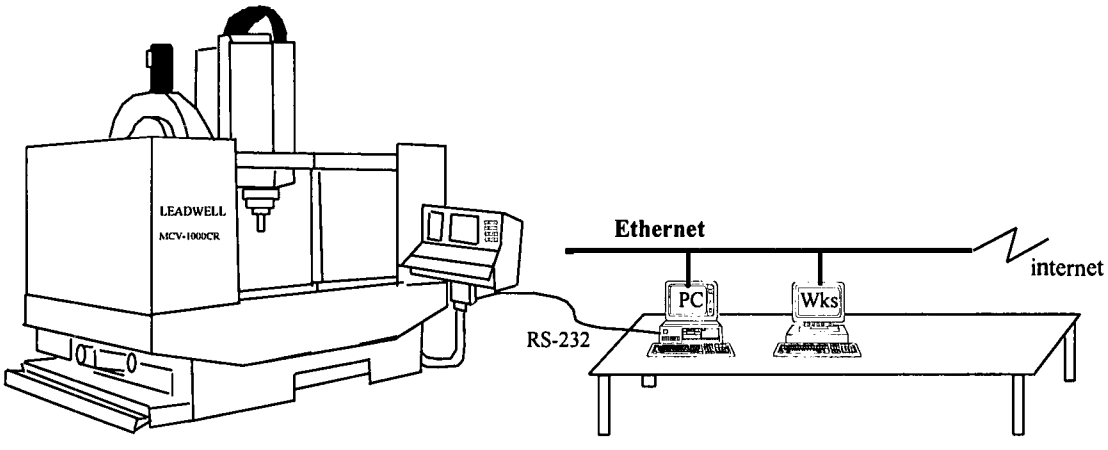


Figure 2.13 The layout of a remote machining unit

2.4 Concluding Remarks

This chapter has presented a computer-supported cooperative environment for CAD collaboration and explored the remote machining functionality. These included the construction of a collaborative CAD based upon existing computer and network resources, the assistance of collaborative tools, and the extension of collaborative CAD/CAM with remote machining facility. Conclusions are summarised as follows.

- The result from the implementation and exploration of the collaborative work showed that the available computer and network environment were not sufficient to support the collaborative work, unless the CAD/CAM application software could have software modules to enhance the communication during the collaborative session.

- The application by incorporating the collaborative tools, for example, shared-application software, showed a very encouraging result. The CAD/CAM software did not need to be changed if there was a shared application software used for the collaboration. However, shared-application software transferred almost all the X-protocols to the other site for the graphics display functions, resulted in heavy network traffic load, and demanded high network bandwidth communication links once the CAD operation became complicated (Hao 1996).

- Owing to the advancement of computing-related hardware and software technology, this chapter has focused on the discussion and presentation in the exploration of enhancing the traditional application software using modern off-the-shelf software products. The utilisation of collaborative and groupware tools: multimedia kits, *ShowMe* Whiteboard, Audio, and SharedApplication, to enhance the distributed designer's communications in fulfilling collaborative design were explained.

- The environment that was explored and established for CAD collaboration was not limited to the application of CAD/CAM. It might be considered as a general purpose environment to wrap application tools for general collaborative work.

- The *ShowMe* Video was not adopted because of the network bandwidth limitation of the Internet.

- The 3D whiteboard (3D geometry viewer) was not adopted because it was not available on the Sun SPARC station platform. The *ShowMe* and *SharedApp* applications could only share the X-Windows application between Sun platform but could not be used between Sun UNIX and PC windows that did not have X-Window installed.

- The data file was kept only on one side, when *SharedApp* was applied, the collaborative session would not be recoverable if the system crashed.

To construct a collaborative working environment for CAD collaboration, a total solution would be the enhancement of the CAD/CAM software in order to use the functions of the data management attributes of the original CAD/CAM tools. Although the data management function in commercial CAD/CAM was not designed for collaborative work, it is expected to be enhanced. In the mean time, the development of a new neutral format such as STEP and OpenGL (OpenGL 1998) format for the CAD/CAM data exchange is expected for 3D whiteboard system or equivalent applications to share 3D views. The Virtual Reality Modeling Language (VRML) and the assistance from the World-Wide Web browsers and 3D CAD geometry viewers can be further explored. However, The current VRML (ISO 1996) does not support collaborative CAD/CAM application on surface modeling and machining. A list of the enabling tools, as shown in Figure 2.14, can be considered to support collaborative CAD/CAM and remote machining.

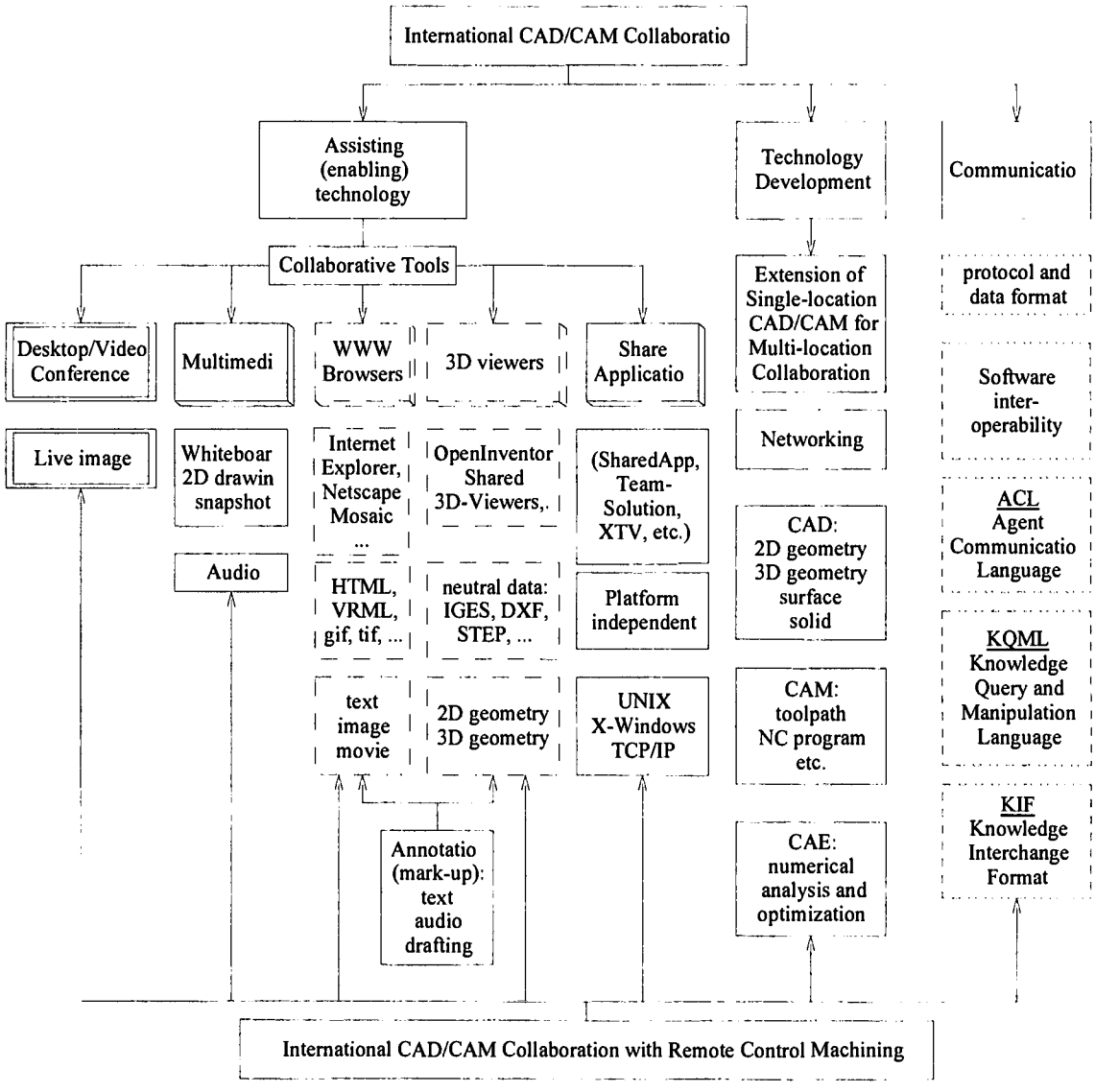


Figure 2.14 Enabling tools for collaborative CAD/CAM and remote machining

Chapter 3 Development of a Remote Machining Cell (RMC)

3.1 Introduction

This chapter presents the development of a remote machining cell (RMC) to extend the international CAD collaboration, as presented in Chapter 2, to an integrated system on the international CAD/CAM collaboration. This development can be used by geographically dispersed designers and manufacturing engineers to perform the collaborative work for the NC milling machining at a distance.

3.2 The Remote Machining Cell

3.2.1 Overview

A remote machining cell must be accessible through the Internet or equivalent network. To achieve this goal, four fundamental modules, which will be discussed in this chapter, were adopted to construct the remote machining cell. These modules include data communication, tool path and NC program generation, workpiece loading and unloading, and unattended NC milling machining. A CNC machining centre was linked to fulfil the remote machining operation for unattended NC machining with loading and unloading tasks operated by an intelligent robot (Lu and Lin 1996). However, the description in the following sections is by no means comprehensive regarding the technical issues, but to show the strategies that were adopted in forming a remote machining cell.

3.2.2 Data Communication

Data files were shared and used for the communication through the network file system (NFS). The remote machining cell consisted of two Sun Sparc workstations (*camr1* and *camr3*), two personal computers (*PCM* and *PCR*), one CNC machining centre controlled by *PCM*, and an industrial robot controlled by *PCR*, as depicted in Figure 3.1. These computers were linked through an Ethernet local area network (LAN). '*camr1*' was used as a World Wide Web (WWW) server controlling manipulation of the incoming message for the remote machining operation. '*camr3*' was used as a server computer for the robot-related messages communication on the loading and unloading tasks assisted by a vision

system and a force/torque sensor (Lu and Lin 1996). The data communication between the CNC machining centre and *PCM* was through the DNC2 interface (Fanuc 1991a), as will be described in Section 3.4.2. The data communication between the robot and *PCR* was through the CommTools (ABB 1992) software library. The communication between the Sun workstations (*camr1* and *camr2*) and the personal computers (*PCM* and *PCR*) was an indirect communication based on the shared data files. These data files were stored in the common database repository that was dedicated for the communication between the personal computers (*PCM* and *PCR*) and Sun workstations. Although the sharing of the file system was a basic function under UNIX system, it was not the case between the operating systems of UNIX and PC. (UNIX operating system could be installed on PC, but the personal computer described in this thesis did not run UNIX. The operating system provided by the PC Windows environment could share directory through the *file manager* and/or the *explorer* but not with UNIX if the UNIX system did not have special NFS server programs installed.) Therefore, to share the file system of the Sun workstation from a PC, a PC-NFS daemon process (a program that was used as a server process to enable the sharing of the network file system) was installed on the Sun workstation. A client process was also installed on the PC as well where the XFS32 (Juhasz 1994) was adopted in order to share the file system from a PC to a Sun workstation. The description of XFS-32 is presented in Section 3.4.3.

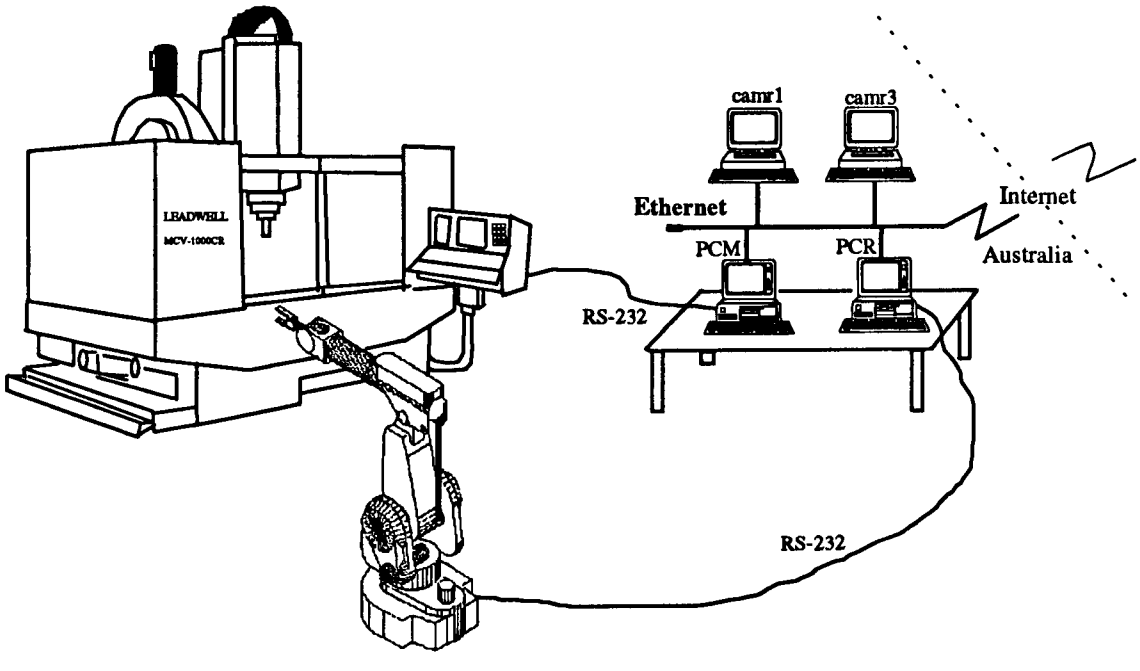


Figure 3.1 The configuration of a remote machining cell

3.2.3 Tool Path and NC Program Generation

The methods that were normally applied to generate the NC program ranged from intensive manual to highly automatic operations (Groover 1987). For example, there were five historically and commonly adopted methods:

1) manual part programming,
2) computer-assisted part programming,
3) manual data input,
4) CAD/CAM assisted NC programming, and
5) computer-automated part programming.

There were two most popular and economical ways (Choi 1994) to create tool path and NC program for computer-aided manufacturing (CAM) systems. These methods were of using 1) a dedicated CAM package and 2) the CAM module of a CAD/CAM software. The toolpath presented in this thesis was generated through the CAM module of a CAD/CAM package in forming an integrated environment for computer-automated programming of NC program based on the given CAD data. The Anvil-5000 CAD/CAM system was adopted for the implementation purpose because of its availability. The graphical application languages *GRAPL-IV* (MCS 1993a) and *Extended GRAPL-IV* (MCS 1993b) were adopted to develop the modules to automatically generate the toolpath and output the associated NC program. Both *GRAPL-IV* and *Extended GRAPL-IV* were the application programming interface (*API*) supported by the adopted CAD/CAM software.

3.2.4 Workpiece Loading and Unloading

The loading and unloading tasks of the workpiece were performed by an intelligent robot (Lin and Lu 1995c & Lu and Lin 1996). The machining centre adopted in the implementation of the remote machining cell was equipped with two automatic clamping devices: air-operated vice and air-operated chuck. The air-operated vice was used as the fixture for three-axis machining operations and the air-operated chuck was used as the fixture for the four-axis machining operation. The workpiece was loaded to the air-operated vice on the table of the machining centre by the intelligent robot for three-axis machining. The robot could automatically search the requested workpiece and perform the loading and unloading tasks assisted by a vision system and a force/torque sensor.

3.2.5 Unattended NC Milling Machining

Unattended machining was one of the fundamental attributes of a flexible manufacturing system for agile manufacturing (O'Connor 1994, Newman 1995) to thrive and prosper in a competitive environment. The layout in the configuration of the remote machining cell, as shown in Figure 3.1, adopted a machining centre controlled by a personal computer *'PCM'* through RS-232 connection for the unattended machining. This configuration was designed to extend the CAD collaboration, as described in Chapter 2, to a physical milling machining operation. To remotely control and monitor the operation of the machining centre, a C program was developed and is described in Section 3.5. The machining centre was *LEADWELL MCV-1000CR* with *FANUC 0MF* controller (Fanuc 1991b). The *DNC2* interface card was installed and was accessible from the personal computer *'PCM'* through serial communication. *DNC2* libraries (Fanuc 1991a) were linked by the Microsoft C program so as to control the machining centre through *DNC2* interface. This program was designed to run in server mode so as to wait for a command requesting for remote machining from the Ethernet network. The command was transmitted through a command file located on the common database repository that was the shared file system in the Sun workstation (*camr1*). The implementation of this remote machining cell is described in Section 3.5.

3.3 System Configuration

3.3.1 Overview

The system configuration of the remote machining cell (*RMC*) is described through two subsections: Hardware and Software. The computer and network facilities in this configuration were linked to show the strategies in forming an integrated *RMC*.

3.3.2 Hardware

The integrated system of the *RMC* consisted of a four-axis (with simultaneously three-axis movement) machining centre *MCV-1000CR*, a personal computer *PCM* which was used to control and monitor the machining centre, a personal computer *PCR* which was used to control a robot, a Sun Sparc station 10 (*camr1*) which was used as the server of the *RMC*, and a second Sun Sparc workstation. The second Sun workstation (*camr3*) was used to link

to *PCM* while the robot's control program was resided in personal computer *PCR*, as shown in Figure 3.1. The *MCV-1000CR* was located at the laboratory of the Centre for Advanced Manufacturing Research (CAMR), University of South Australia, Australia.

3.3.3 Software

The software in this integrated system consisted of home-brew programs, public domain shareware, and off-the-shelf commercial products. These software included:

1) a commercial CAD/CAM software Anvil-5000 and its graphical application languages *GRAPL-IV* and *Extended GRAPL-IV,*

2) a C program that ran on PCM and acts as a controller and monitor of the machining centre *MCV-1000CR*,

3) a C program ran on PCR to control the robot,

4) a C program resided in *camr3* for image processing serving as the vision system for the robot to automatically recognise the requested workpiece, and

5) the programs to transmit the status of the robot and the machining centre audiologically in enhancing the user interface in checking the status of RMC.

The source codes of the control and monitor programs of the machining centre are listed in Appendix B for reference.

3.3.4 The Configuration

This section describes the system configuration of the remote machining cell, as shown in Figure 3.2, which adopted the XFS-32 to share (mount) Sun Sparc workstation's hard disk resources (common database repository) for the data transmission. The data communication among the control programs was achieved through data files that were located on *camr1* and *camr3*. These files were shared by *PCM* (the controller of the machining centre) and *PCR* (the controller of the intelligent robot), respectively. The

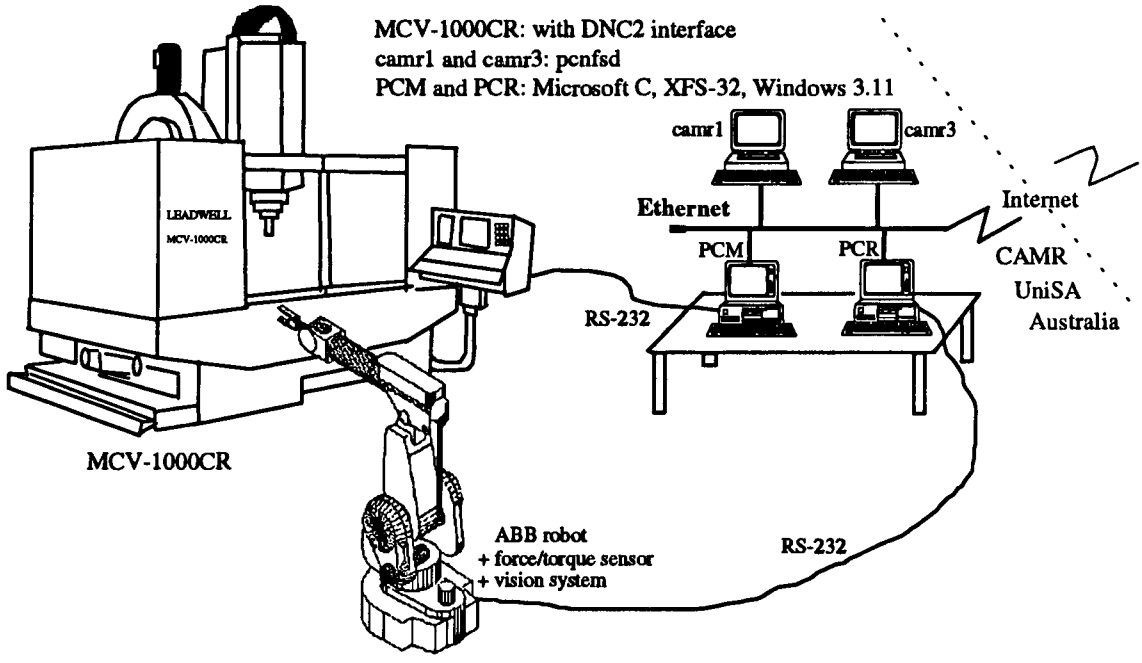


Figure 3.2 The system configuration of the remote machining cell

system configuration, as shown in Figure 3.2, is an extension of Figure 3.1. Both *PCM* and *PCR* were installed with Microsoft C programming language compiler, XFS-32 and Windows 3.11 operating system. The machining centre *MCV-1000CR* was equipped with DNC2 interface card which was controlled by *PCM* through RS-232 serial port. *'camr1'* was equipped with an audio speaker to output audiologically the status of the remote machining cell (Kao and Lin 1997a). This was achieved through C programs playing the pre-assigned audio files adapting different occasions and also through the World-Wide Web navigator that can interpret JAVA application (Gosling 1996). The system communication adopted in this integrated system is described in Section 3.4.

3.4 Description of the RMC System Communication

3.4.1 Overview

This section describes the system communication strategies that were adopted to link different hardware devices in the remote machining cell. These include the DNC2 interface, XFS-32 client program to share the network file system, and the network file system (NFS). Exchange data via the files residing in the server computer implies a logical star topology underlying an Ethernet local area network.

3.4.2 The Description of DNC2

The DNC2 interface was a serial communication protocol between the FANUC CNC unit and a personal computer through the RS-232 interface (Fanuc 1991a) with a three-layer structure. This structure included the application layer, the data link layer and the physical layer, as shown in Figure 3.3. The communication between a CNC and a personal computer could be established based on the communication protocols of these layers. These communication protocols were used for one-to-one (point to point) communication. Serial communication was the process of converting an 8-bit byte to a stream of 8 single data bits travelling consecutively through a single wire according to the RS-232 communication protocols. The RS-232 communication protocols were developed by the Electronic Industries Association and officially known as the EIA interface standard RS-232c. The connection point for RS-232 serial communication was called a serial port and the serial port of a PC was called the COM port. Each COM port was a separate serial port that could be used for RS-232 connections.

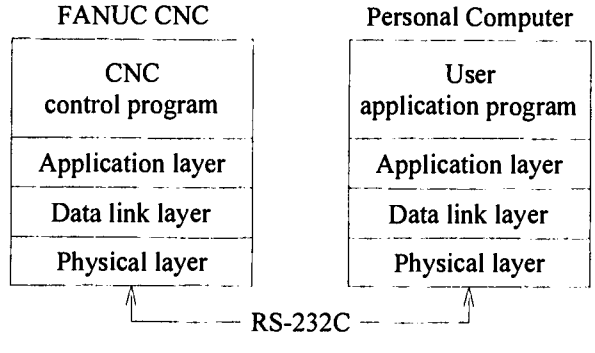


Figure 3.3 Three-layer structure of DNC2

There were two parts in the application of the DNC2 communication. One was a hardware interface card which was installed inside the CNC control cabinet of the machining centre. The other was a Microsoft C communication program library in DOS format that could be linked through a Microsoft C program on PC. This library consisted of forty-four callable subroutines, as listed in Table 3.1 (Fanuc 1991a).

Table 3.1 DNC2 subroutines

No.	Function	Description
1	D_download	Send NC program (from PC to CNC)
2	D_download_wait	Waits for download request and starts downloading a NC program
3	D_upload	Get NC program (from CNC to PC)
4	D_delete	Deletes NC program that resides in CNC controller
5	D_fdir	Checks if NC program exists in CNC
6	D_alldir	Stores all NC program numbers in CNC to a file
7	D_alldir_mem	Stores all NC program numbers in buffer
8	D_freemem	Reads the size of free memory in CNC
9	D_tool_pos	Reads the tool position
10	D_servo	Reads the servo delay
11	D_AccDec	Reads the acceleration/deceleration delay
12	D_diagnose	Reads the machine interface signal data
13	D_exec_num	Reads the number of program or block being executed by the CNC
14	D_parameter_r	Reads the CNC parameters
15	D_parameter_w	Writes the parameters stored in a file to the CNC
16	D_parameter_w_mem	Writes the parameters stored in buffer to the CNC
17	D_pitch_r	Reads pitch error data and saves into a file
18	D_pitch_r_mem	Reads pitch error data and stores into buffer
19	D_pitch_w	Write pitch error data stored in a file to CNC
20	D_pitch_w_mem	Writes pitch error data stored in buffer to CNC
21	D_offset_r	Reads tool offset data and stores the data in a file
22	D_offset_r_mem	Reads tool offset data and stores the data in buffer
23	D_offset_w	Writes the offsets data in a file to the CNC
24	D_offset_w_mem	Writes the offset data in buffer to the CNC
25	D_variable_r	Reads custom macro variables and stores the values into a file
26	D_variable_r_mem	Reads custom macro variables and stores the values in buffer
27	D_variable_w	Writes the macro data stored in a file to CNC
28	D_variable_w_mem	Writes the macro data stored in buffer to CNC
29	D_toollife	Reads tool life management data in CNC and stores into a file
30	D_toollife_mem	Reads tool life management data in CNC and stores in buffer
31	D_modal	Reads automatic operation modal information and save into a file
32	D_modal_mem	Reads automatic operation modal information and stores in buffer
33	D_actual_speed	Reads actual speed of specified axes into buffer
34	D_ADconvert	Reads A/D conversion data
35	D_alarm	Reads alarm information
36	D_status	Reads status information
37	D_message	Output message to CNC
38	D_PMCdata_r	Reads PMC data (for handshaking to link with D_PMCdata_w)
39	D_PMCdata_w	Writes PMC data (for handshaking to link with D_PMCdata_r)
40	D_select	Asks CNC to search for a NC program for D_start DNC operation
41	D_start	Cycle start a NC program
42	D_reset	Directs the CNC to reset the system
43	D_systemID	Reads system ID: CNC model name and DNC2 software version.
44	D_initial	Initialises the specified communication port and protocols

These subroutines were selectively used, as shown in Figure 3.4, to control and monitor the CNC machining centre according to the algorithm developed, as will be explained in Section 3.5, to operate the machining centre remotely. The subroutine *'D_initial'* needed to be actuated in order to set and to initialise the communication protocols to link the PC and the machining centre through the DNC2 interface. The communication speed was set to 4800 bits per second, the data length was 8 bits with none parity and one stop bit, and the protocol was in ASCII format. The subroutine *'D_fdir'* was then used to check whether there was an NC program in the CNC controller with the same program number as the NC program to be downloaded to the CNC controller. If this was the case, the subroutine *'D_delete'* would be used to delete the NC program. Otherwise, the NC program would then be downloaded from the PC to the memory of the CNC controller by the subroutine *'D_download'*. The subroutine *'D_select'* was used to obtain the NC program in the CNC controller. The subroutine *'D_start'* was used to *'CYCLE START'* this NC program for the milling operation. Once the milling operation was started, the subroutine *'D_PMCdata_r'* was used to monitor the status of the machining centre. The status of the machining centre was monitored by checking the diagnostic parameter number *'F0148'* (Leadwell 1993). The letter *'F'* here refers to the signal from the CNC (mother board of the FANUC controller) to the PMC (programmable machine controller). There were 8 bits in the parameter of *F0148* (from bits 0 to 7). Bits 0 to 3 represented the status of zero return positions for the axes X, Y, Z, and A, respectively. Bits 5 and 7, represented the machining status. If both bits 5 and 7 were equal to one, the machining centre was still in the operation of machining. The control and monitor program residing in the PC could not accept any new machining command while the machining centre was still in the milling operation. If both bits 5 and 7 were equal to zero, the status of the machining centre was idle and ready for receiving another message for machining operation. The subroutine *'D_alarm'* was used to check if there was any alarm message from the machining centre while the machining centre was conducting the milling operation remotely. The counterpart of DNC2 settings for the parameters in the CNC controller is listed in Table 3.2 where *'x'* represents keeping the original setting. The connection on the physical layer of the RS-232 communication between the CNC and a PC is shown in Figure 3.5.

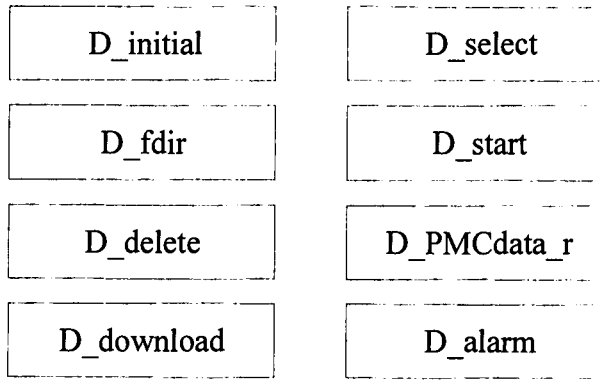


Figure 3.4 The selected DNC2 subroutines

Table 3.2 The parameter settings in CNC controller for DNC2 operation

Parameters	Settings
0051	xx010000
0055	xxxx0xxx
0251	10
0365	10
0366	10
0368	10
0369	10
0370	10
0396	xxxx1000
0490	80
I/O	10

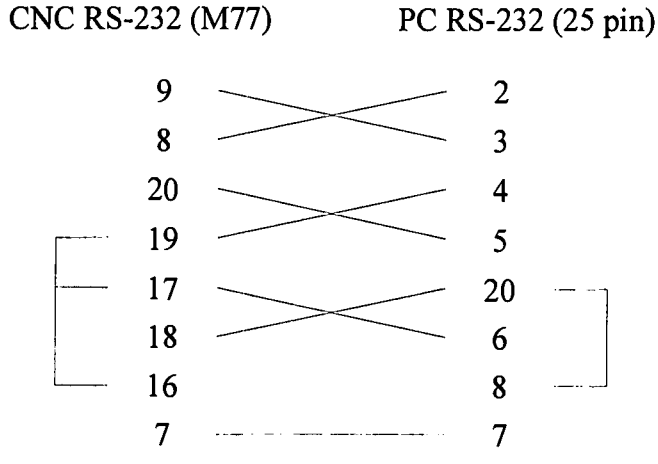


Figure 3.5 RS-232 connection between CNC and PC for DNC2

3.4.3 The Description of XFS-32

The *XFS-32* was a shareware in public domain and needed to be installed with TCP32. This software could be found through the *archie* network service (Goldstein 1991) or *World-Wide-Web* search engines. *Archie* maintained a list of archive sites on the Internet which provided an anonymous FTP service. The archie server was '*archie.au*' in Australian Academic and Research Network (AARNet). The file names of the shareware were '*TCP32.ZIP*' for TCP32 and '*XFS32120.ZIP*' for *XFS-32*. Users could have the capability, through *XFS32*, to share the hard disk resources of a workstation which runs PC-NFS daemon. *XFS32* was a client program implemented to share Network File System (NFS) for PC's running Microsoft Windows for Workgroups 3.11 and Microsoft TCP/IP-32. It used the network-wide file sharing of other systems and provided the enormous disk resources of computer workstations. The file sharing mechanism was transparent so that the traditional PC software such as file managers and word processors could be used to manipulate remote files. Therefore, the files to be shared and used for the control message passing could be located in the exported directory under UNIX environment. The XFS32 that was adopted in this research was version 1.0 by Robert Juhasz (Juhasz 1994).

Microsoft TCP/IP-32 for Windows for Workgroups version 3.11 was a 32-bit implementation of the industry-standard Transmission Control Protocol/Internet Protocol (TCP/IP). The TCP/IP protocol was a networking protocol that provided communication across interconnected networks made up of heterogeneous computers with diverse hardware architectures and various operating systems. The TCP/IP could be used with Windows for Workgroups alone, to communicate with devices using other Microsoft networking products, or to communicate with non-Microsoft systems, such as UNIX. The TCP/IP protocol family was a standard set of networking protocols regarding how data were passed between computers on a network. The TCP/IP was used to connect the Internet, the worldwide internetwork connecting over millions of sites. (By convention, the first character of the word Internet is capitalised when it is referred to the worldwide internetwork in this thesis.) The TCP/IP protocol was supported by most of the modern operating systems. Most of large networks in the world relied on TCP/IP for much of their network transaction. The Microsoft TCP/IP-32 provided a robust, scaleable, and cross-platform client-server framework for network connection and supported the XFS-32 client program.

3.4.4 The Description of Network File System (NFS)

The network file system (NFS) was a network protocol that allowed a client computer (such as a workstation) to remotely access disk files on a server computer (such as a file server). NFS was created at Sun Microsystems (Callaghan 1996) in the early 1980's. The NFS provided a transparent method of transmitting and receiving files across a machine architecture and operating system independent (heterogeneous) network (Dataquest 1995). The system manager could mount (attach) the directory structures (file systems) of a remote computer onto the local file system (Sun 1990 & MCS 1993c). Although the mounted file systems were not physically attached, they behaved as though they were an extension of the local directory structure. A new directory path to files which in reality resided in another system was then provided by NFS in a way of totally network transparent to the users. There were eighteen NFS operations which could be viewed as the primitive operations upon which all other file operations could be built. NFS operations were implemented using a facility known as RPC (Remote Procedure Call). A feature of RPC was that a client made a request and the server replied once the operation was completed (Cousins 1992). (The client would never trust the server if a request was not matched by a reply within a short period of time, the client was responsible for retrying the operation.) NFS allowed users to easily share files between a PC and a UNIX workstation for sharing the heterogeneous information.

3.5 *Integration of the RMC System*

This section describes the implementation methodology in fulfilling the integration of the remote machining cell. The remote machining cell (*RMC*) consists of three modules (processes), as shown in Figure 3.6, linking the collaborative CAD/CAM (*COCADCAM*) system which will be presented in Chapter 4. The first module is a control and monitoring process that resides in a personal computer *PCM*. This module controls and monitors a machining centre which links to a robot for the loading and unloading tasks of the workpiece. The second module is a status updating process that resides in the server computer (Sun workstation in this implementation) playing the role to transmit audiologically (playing a recorded audio file) the updated status of the remote machining cell. This is to enhance the status awareness of the *RMC* that is operable at a distance. The third module is a robot control program playing the role to control a robot for the loading and unloading tasks. These modules communicate with each other through message files underlying a shared network file system, as described in Section 3.4.4.

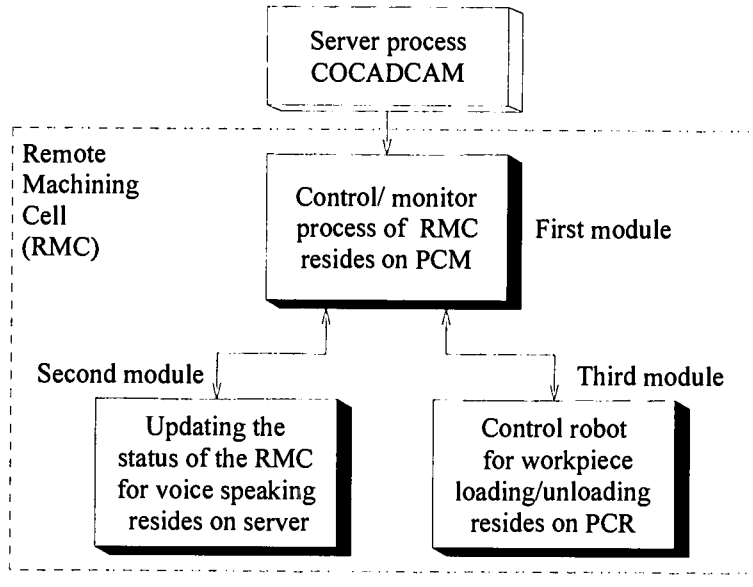


Figure 3.6 Relationship between RMC and *COCADCAM*

There are four sub-modules in the first module, as shown in Figure 3.7, besides the main module. These sub-modules are:

1) Robot loading,
2) 2) Download of the NC program,
3) Monitoring of the machining operation, and

4) Robot unloading.

The main module consists of all the functions that do not belong to these modules. The main module listens to the command message through a shared command file that resides in the server computer and is to be output by the server process *COCADCAM*, as will be explained in Section 4.2. The main module parses the messages, reports the status of and operated the remote machining cell as is requested by the command messages. The first sub-module *'Robot loading'* will be actuated if the command message that is received by the main module includes the link to a robot asking for workpiece loading and/or unloading. If there is a link to robot, the first sub-module will open the door of the machining centre, unclamp the fixture and send out message through a shared file to notify and ask robot for the loading task. A command message will be sent out to the second module once the door of the machining centre is opened, and an audio file 'The door is opened' will be played. An audio file *'The vice is unclamped'* (or *'The chuck is unclamped'*) will be played as soon as the fixture (vice or chuck) is unclamped. Other audio files *'Robot is loading the workpiece'*, *'Robot is still loading'* or *'Robot loading is completed'* will be played consecutively whenever appropriate. The first sub-module will be notified once the loading process of the workpiece is finished.

The first sub-module will then clamp the fixture (vice or chuck) and play the audio file *'The vice (or chuck) is clamped'* after it receives the message saying the loading task is finished.

The second sub-module, as shown in Figure 3.7, is then actuated to download the NC program into the controller of the machining centre. If there is already one NC program with the same program number existed, for example O1000, in the controller, this sub-module will delete the NC program. The requested NC program will be downloaded if the memory in the controller is large enough, otherwise an alarm will be alerted. The requested NC program will then be started (*CYCLE START*) for the milling operation.

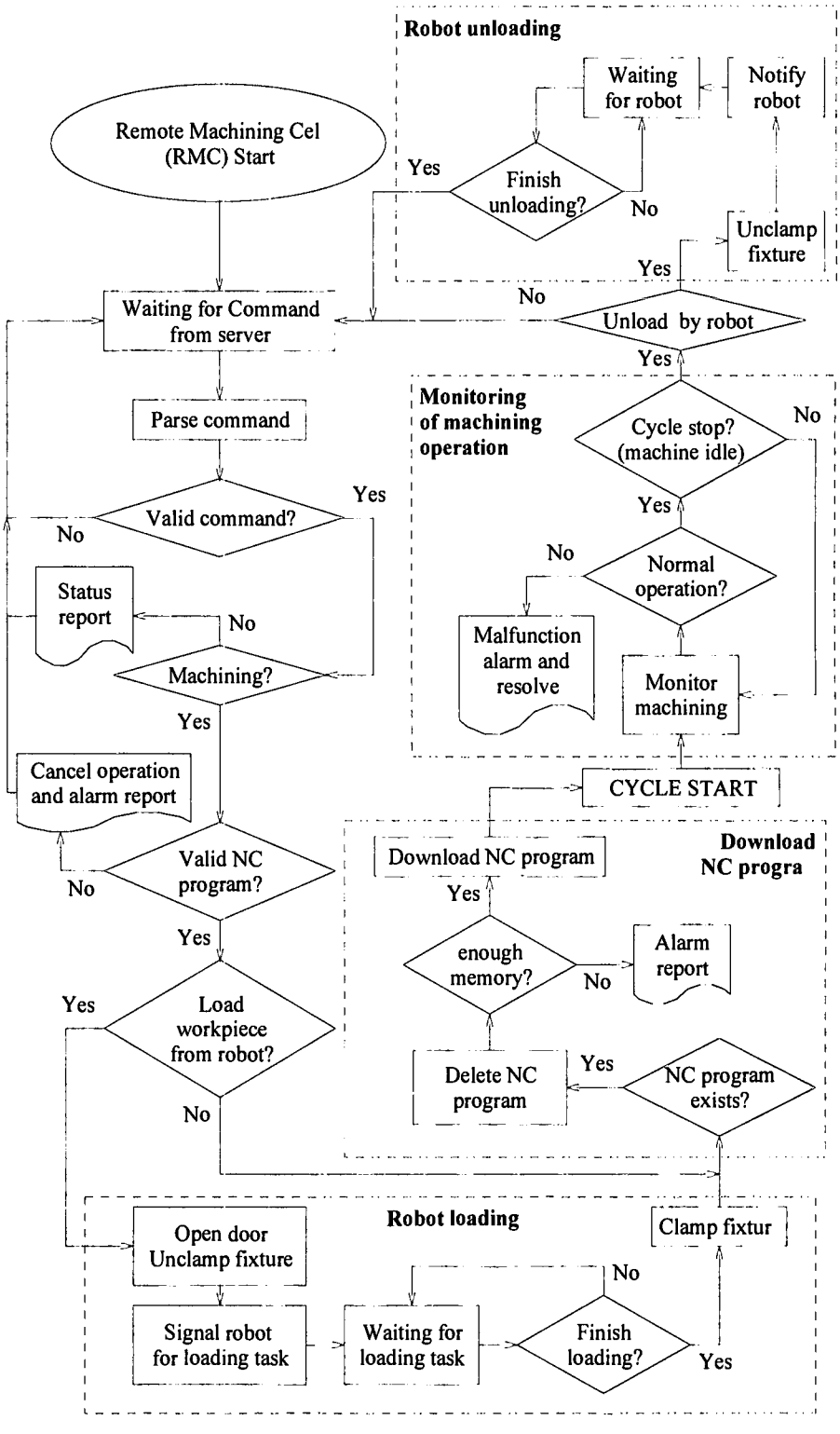


Figure 3.7 Flowchart of the RMC control and monitor process

The third sub-module will then start to monitor the machining operation by enquiring the CNC controller regularly. This communication is achieved through the DNC2 interface. If there is an abnormal operation, a malfunction alarm will be reported to notify the remote operator to resolve the problem. If the machining operation is normal, that is no alarm message, this module will check regularly whether the machining operation is finished (*CYCLE STOP*) or not (Section 3.4.2). The monitoring process will not stop until either the machining process is finished or alarm occurs. The status of the machining centre can be transmitted audiologically by playing the recorded audio files during the operation.

The fourth sub-module is 'Robot unloading' which will be actuated by the third module once the machining operation is finished (*CYCLE STOP*) and the command message requests the unloading task of the workpiece by robot. This sub-module is very similar to the first sub-module 'Robot loading' except unloading rather than loading task is performed. This sub-module will unclamp the fixture, notify robot to perform the unloading task, and then return to the main module in the status of 'waiting for command from server'. The source codes in implementing these modules are listed in Appendix B.

3.6 Concluding Remarks

- This chapter has presented the development of a remotely operable machining cell in extending the collaborative CAD, as described in Chapter 2, to a collaborative CAD/CAM (*COCADCAM*) application. The linkage between *COCADCAM* and the remote machining cell will be illustrated in section 4.4.4.

- The status of the remote machining cell was reported regularly and the status of the cell was transmitted audiologically to ease the monitoring process of the remote machining cell.

- The robot server for workpiece loading and unloading was developed to shorten the setup time of the workpiece; this robot server consisted of a robot with a vision system and a force-torque sensor (Lu and Lin 1996). The implementation of the intelligent robot server enabled a remote machining cell, as depicted in Figure 3.2, that could be operated at a distance through the Internet.

- The remote machining cell could be used for production services with rapid response to the machining task from the Internet (Park 1993) and could also be accessed through the World Wide Web (Kao and Lin 1997a).

- The remote machining cell was a virtual machining cell which was a system consisted of software modules that operated a physical machining unit (Raulefs 1994).

- A flexible fixture or universal fixture would also be necessary. However, instead of creating a fixture that could be used to adapt all shapes of workpieces, a flexible fixture device that is adjustable to adapt to various types of fixturing would be preferred. For example, vice could be used for clamping rectangular raw materials.

- Coordinate measurement machine (CMM) could be used for the after-process inspection. The measured data should have feedback to the machining centre for the modification of the machining operation (Chen 1990).

- This system has been successfully applied to: 1) the linkage to a collaborative CAD/CAM system (*COCADCAM*) serving a facility for machining the co-edited 3D NURB free form geometry collaboratively once the NC program is generated, and 2) a remote machining operation for a forward extrusion die electrode surface. Examples for the demonstration will be presented in Chapter 5.

Chapter 4 *COCADCAM* -- a Collaborative CAD/CAM System

4.1 Introductory Comments

This Chapter presents the development of a collaborative CAD/CAM system (*COCADCAM*). *COCADCAM* can be used for a collaborative work linking two geographically dispersed CAD/CAM engineers working on a CAD/CAM based task such as the co-editing of three-dimensional CAD and derived data for remote machining.

The *COCADCAM* system extends the application of single-location CAD/CAM technology to multi-location technology for geographically dispersed CAD/CAM collaboration, for example, in the fulfillment of collaboratively and interactively editing CAD geometry at a distance. This system adopts the UNIX interprocess communication (*IPC*) and a socket-based connection-oriented client and server model for data communication. *COCADCAM* can output a numerical control (NC) program following a CAD-geometry coediting session and facilitate a collaborative design and manufacturing system. Geographically dispersed locations can be linked to form a virtual team underpinned by a single-location CAD/CAM environment through the Internet or the Integrated Services Digital Network (ISDN) for CAD-geometry coediting and remote machining. Thus, a new product that is designed on one side of the world can be machined on the other side by collaborative CAD/CAM users. Examples will be used to explain *COCADCAM* for multi-location collaboration such as the CAD refinement at the final stage of a design task.

The motivation behind this research was to develop an algorithm allowing geographically dispersed CAD/CAM engineers to work collaboratively and interactively as a concurrent team on three-dimensional CAD-geometry related tasks. The *COCADCAM* system was then implemented to offer the distributed engineers a new working environment by linking network functions to a single-location CAD/CAM system for collaborative work on CAD data, forming a virtual multi-location CAD/CAM application. The network response time for data transaction between the cooperating users was shortened by using the algorithm proposed in this research which transmitted only the necessary data. This development also explored the data transaction time through a local area network (LAN), the Internet and the ISDN.

4.2 The COCADCAM System

4.2.1 Overview

The development of *COCADCAM* was based on the system structure of a conventional single-location CAD/CAM software which only supported individual work and isolated one another (Eltoweissy 1993). *COCADCAM* extended a single-location software to a multi-location application by

(1) proposing new data interchange formats because the general neutral CAD interchange formats were not defined for dynamic CAD-geometry data exchange, and

(2) implementing a communication and networking algorithm on top of the protocol suite of TCP/IP to form a platform independent working environment.

The TCP/IP protocol suite supported a reliable connection-oriented service between application programs. This service was also termed a virtual circuit since it appeared to have a dedicated connection between the application programs. This connection was similar to a telephone call which established a link with the other party, exchanged data (conversation) and then closed the connection. CAD/CAM users at different locations could interact and collaborate on the CAD-geometry data through *COCADCAM* and generate a NC program on a collaboratively discussed and agreed free-form surface for the milling machining operations.

There were two subsystems in *COCADCAM* for CAD/CAM functions: 1) a collaborative CAD subsystem (*COCAD*) and 2) a collaborative CAM subsystem (*COMAN*). The system structure of *COCADCAM* is illustrated in Figure 4.1. *COCAD* supported CAD/CAM engineers to co-edit three-dimensional CAD geometry and the *COMAN* subsystem supported geographically distributed remote manufacture. The *COMAN* subsystem was also linked with a remote machining cell which could be operated through the Internet and the ISDN following the collaborative work. The functions of *COCAD* was developed based on the application programming interface (API) of a CAD/CAM system; *COCAD* also controlled the network data interpretation, geometry construction and modification. *COMAN* controlled the generation of a toolpath and the postprocessing of an NC program towards the collaborative manufacturing in machining.

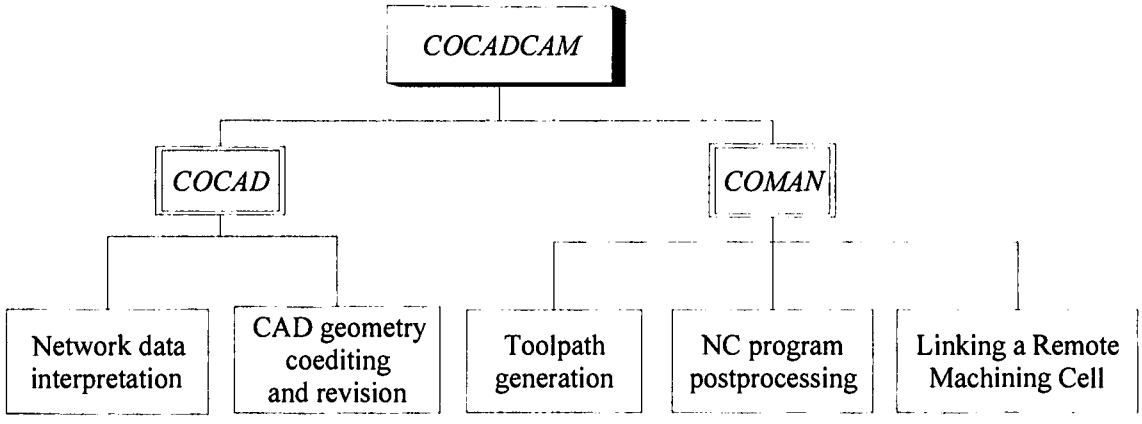


Figure 4.1 System structure of *COCADCAM*

Both *COCAD* and *COMAN* linked the networking algorithms to connect two geographically dispersed *COCADCAM* users.

4.2.2 The System Configuration of *COCADCAM*

The configuration of the *COCADCAM* system proposed in this research is shown in Figure 4.2. The database system in *COCADCAM* was a hybrid of a central architecture and replicated architecture (Shu 1992) so that every collaborative designer kept a copy of the file that was under discussion. '*gserv*' was a global server controlling the token passing. The user (designer) must have the *token* before he or she could create or manipulate the geometry that was to be co-worked. The geometry included CAD geometry and a derived entity such as toolpath. The user was either named '*creator*' while holding the *token* or

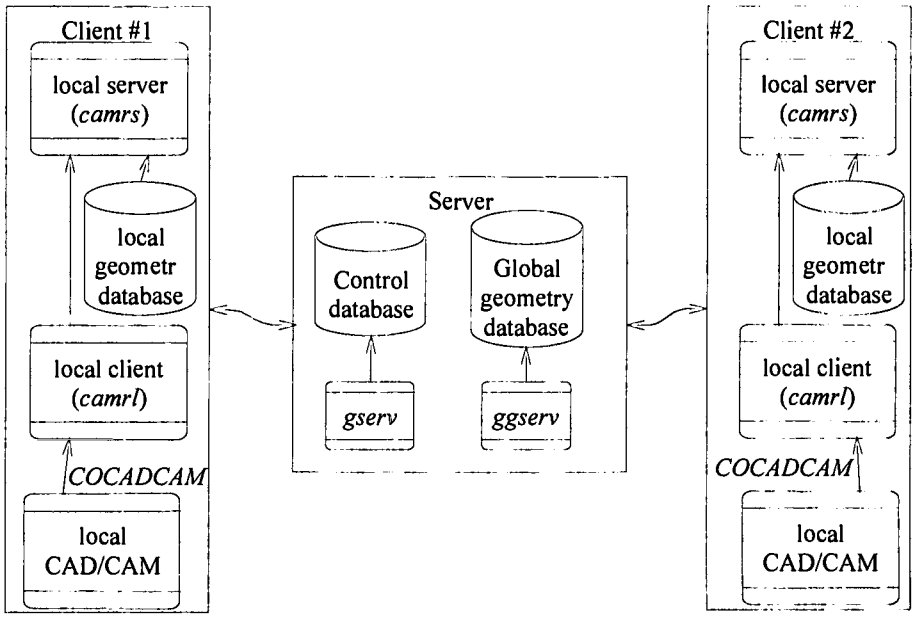


Figure 4.2 System configuration of *COCADCAM*

'observer' while not holding the *token*. *'ggserv' was* a global geometry server controlling the geometry data passing and communication. Both *gserv* and *ggserv* were always observing and listening to the incoming messages through pre-assigned well-known TCP ports on the server computer. *'camrs'* was a local server that controlled the communication between the local client *'camrl'* and both *gserv* and *ggserv* servers. The local client *camrl* linked up a CAD/CAM system with *camrs*. All the server programs *gserv*, *ggserv*, and *camrs* were concurrent servers that were developed in the C programming language. A concurrent server was a computer process that was always waiting for the contact by a client process and to serve the client process in a concurrent fashion (Stevens 1994). A process was a piece of computer program that was being executed and controlled by the computer's operating system. The local client process *camrl* was also written in the C language. *COCAD* linked up a CAD/CAM software and the local client *'camrl'* through the API, for example, the graphical programming languages *GRAPL-IV* and *Extended GRAPL-IV* (MCS 1993a and 1993b) of Anvil-5000 CAD/CAM software. The API provided the tools that were needed for users to develop their applications to interface and to directly access the CAD/CAM data base. The UNIX interprocess communication (IPC) was used for the data communication between *COCAD* and *camrl*. The role of a client was either *creator* or *observer*. The *creator* was the first user who asked for the *token* from global server *gserv*. The *observer* was the one who did not have the *token*. The *token* could be dropped whenever the *creator* chose the *'drop token'* function. A socket interface (Frost 1994) was used for the intra- and inter-network communication. The socket interface was used to connect one process to the other. In a typical socket connection flow of a connection-oriented communication, as shown in Figure 4.3, the server process listened to the incoming requests through a well-known port, which forked (duplicated) itself to manipulate the requests and listened again. The forked server process, which manipulated requests and interacted with the client process, was ended once the interactive session was concluded by the client process. Although a commercial CAD/CAM system was adopted in developing *COCADCAM*, it was not intended to promoting this software but only for the implementation purpose to construct a collaborative CAD/CAM system by using a system that had both CAD and CAM functions supported by its API.

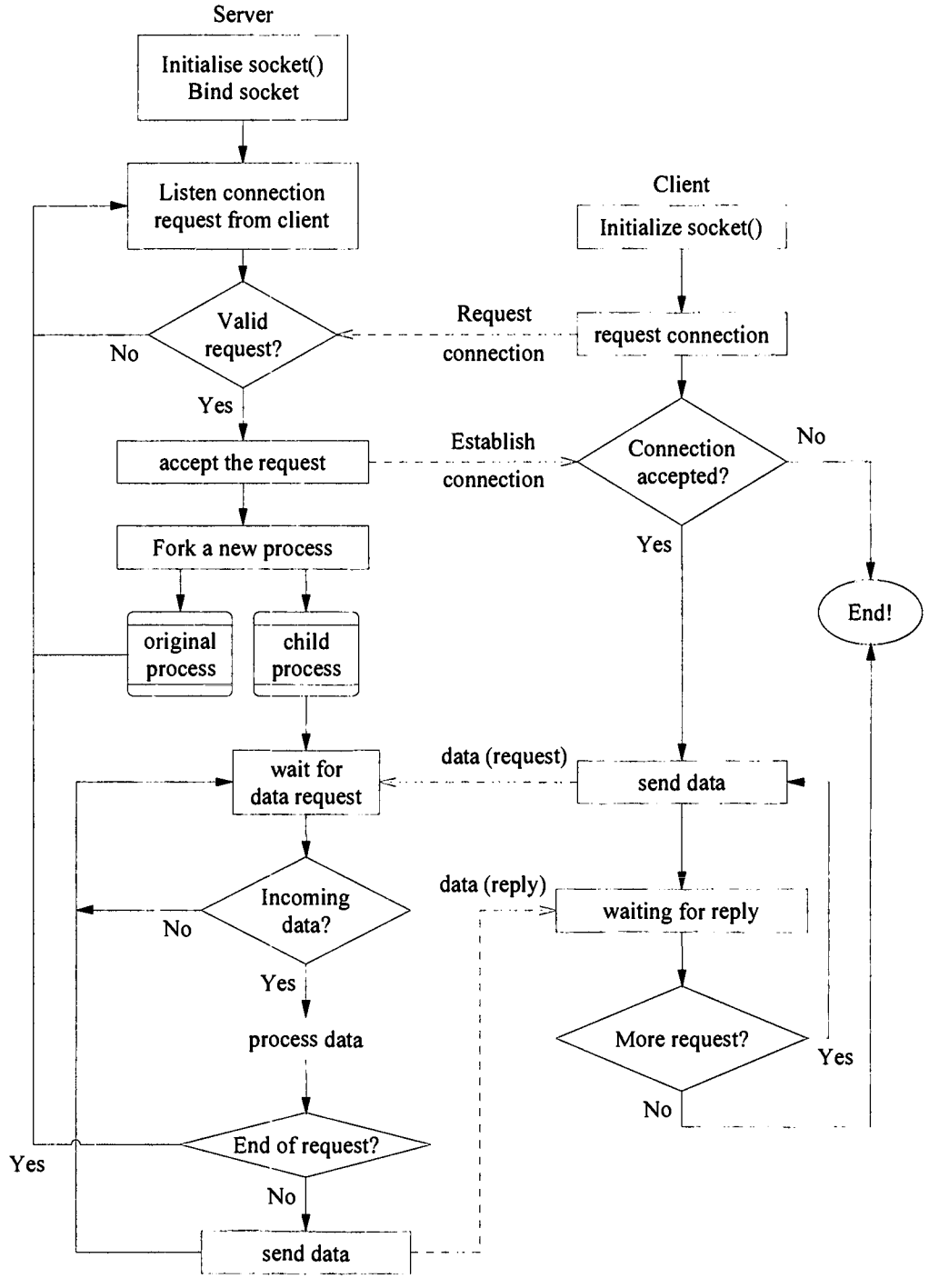


Figure 4.3 Socket connection flow for connection-oriented client-server model

The algorithm of a connection-oriented concurrent service is illustrated in Figures 4.4, 4.5 and 4.6 (Stevens 1994). Let us assume a concurrent server is started on a host computer, called *CAMR1*, as shown at the left-hand side of Figure 4.4, and a communication channel is opened, by using a well-known port (6100, for this example), to wait for a client request. The notation {TCP, *, 6100} is also assumed to represent a server that is waiting for a TCP connection from the network on port 6100. If a client starts up and requests for the connection to the server *CAMR1* from another computer, called *CAMR2*, as shown at the right-hand side of Figure 4.4, the connection can be shown as in Figure 4.4. The ephemeral port number for the client application is assigned by *CAMR2* (3000, for this example). The notation {TCP, *CAMR2*, 3000} is called the client's socket. When the server process receives the client's request for connection, it forks a copy of itself as a child process, passes the connection to the child process, as shown in Figure 4.5, and returns to its waiting loop. The child process then handles the client's request and forms an association {TCP, *CAMR1*, 6100, *CAMR2*, 3000} to distinguish this connection from the other connection. If there is a connection request from another client process on the same host *CAMR2* again to the same server *CAMR1*, another ephemeral port number, say 3100, would be assigned by *CAMR2* as a unique number to the new client process. That is to say a new association of {TCP, *CAMR1*, 6100, *CAMR2*, 3100} would be formed for the new connection. The connection would then look like Figure 4.6 with two associations formed and the server process is still available for another new connection request.

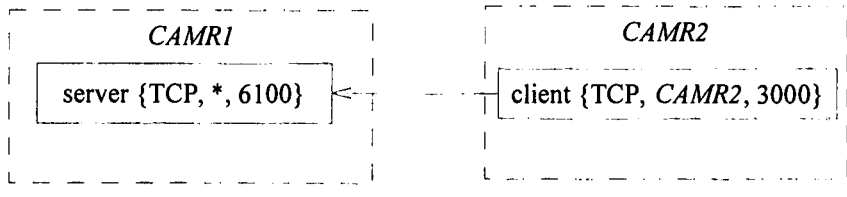


Figure 4.4 A sample connection from client to server

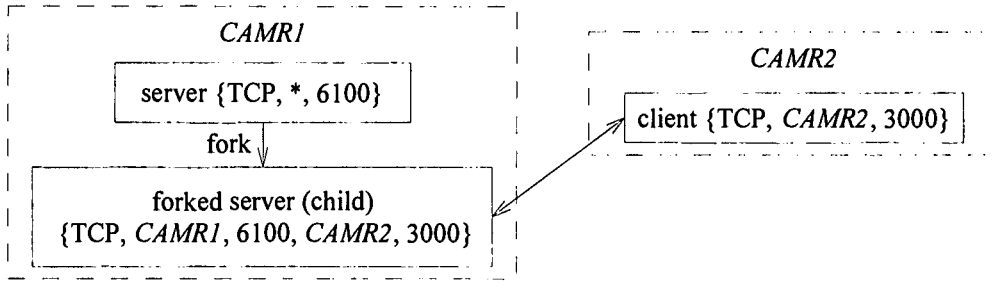


Figure 4.5 A concurrent server passes connection to a forked child process

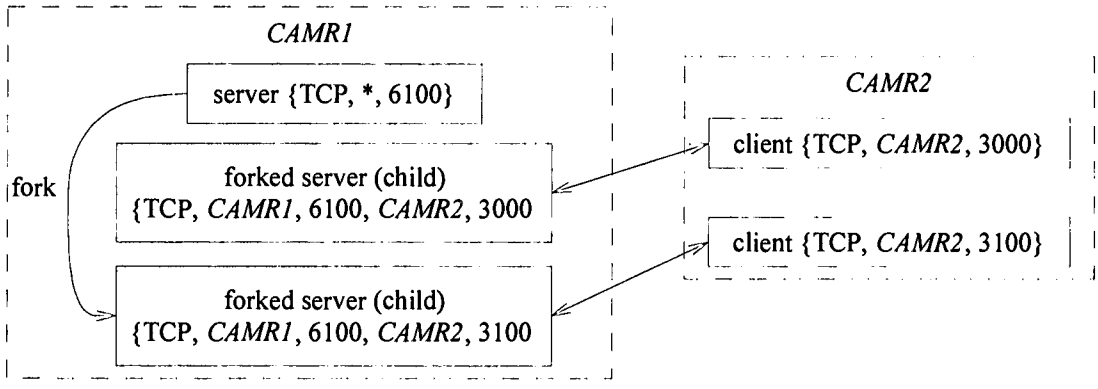


Figure 4.6 The concurrent server passes connection to another forked child process

The layout of *COCADCAM* is depicted in Figure 4.7 where SUN Sparc station 10s were used as the major network nodes. Every node in this system was linked through the Ethernet campus local area network. The server node could be the same as either *'Client #1'* or *'Client #2'*. A CAD/CAM software was ran by the clients, on SUN Sparc workstations, as soon as the collaborative CAD/CAM system *COCADCAM* started. The global data base (the common database on server node) stored all the design history and data generated from the collaborative session. The global data base was the central database that was used by the global geometry server *ggserv*. The network configuration of *COCADCAM* also supported data communication among different application programs including server, client, and CAD/CAM software. Examples of synchronous, dynamic and interactive collaboration between geographically dispersed *COCADCAM* users will be presented in Chapter 5.

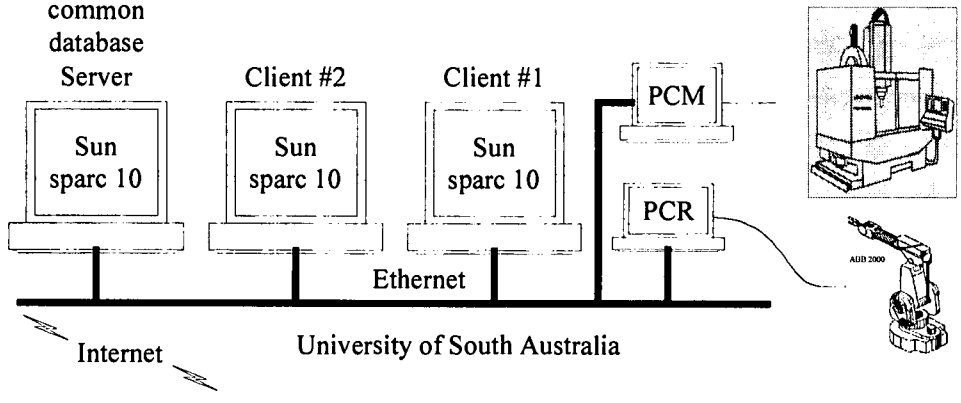


Figure 4.7 Layout of the physical devices in *COCADCAM*

4.2.3 The *COCADCAM* Program Structure

The program structure of the *COCADCAM* system is shown in Figures 4.8, 4.9 and 4.10. There are nine major functions in *COCADCAM*; functions 1-5 are illustrated in Figure 4.8, and functions 6-9 are illustrated in Figure 4.9. Sub-functions that are not shown in Figure 4.8 (sub-functions 3.3, 3.4 and 4.3) and Figure 4.9 (sub-function 7.3) are depicted in Figure 4.10. The *COCADCAM* system is a menu-driven system based on the API of the adopted CAD/CAM software. The global server programs must be initiated on the server computer and the local server program must also be up and running on the computer of every collaborator before the *COCADCAM* system can be started. A sample layout is illustrated in Figure 5.20(a) where the global server (*gserv*), the global geometry server (*ggserv*) and the local server (*camrs*) coexist on the same computer. Furthermore, Figure 5.20(b) depicts the other local server (*camrs*) running on another computer.

The first function is "Access Control", as shown in Figure 4.8. A user cannot create new or edit old geometry until the user holds the token through "Whose Control" sub-function; the user will remain observing the token-holder's operation for collaborative work if he or she does not hold the token. The *creator* is the user who has taken the control (holds the token), otherwise the user is an *observer*. The *observer* will keep on observing the activity of the *creator* until the *creator* issues "Pause Drawing" to release the *observer*. A user who issues "Whose Control" will automatically be offered the control token by the global server (*gserv*) and become the *creator* if no one has taken the control or the last *creator* has dropped the control. The control token can be dropped by the *creator* by issuing the "Drop Control" sub-function.

The second function "Entity Control" is used to manage the existed CAD/CAM geometry and entity through their specific entity names. The third function "Geometry Construction" allows the *creator* to create a new geometry such as an interpolated NURB curve (by selecting the "Interpolated" from "NURB splines" of sub-function 3.3 in Figure 4.9). When a user, who does not have the control token, clicks the third function "Geometry Construction", the user will be asked if he or she wants to proceed to become the *observer* to observe the *creator's* action. If the answer "Yes" is selected, the user (now the *observer*) will keep on observing until the *creator* issues "Pause Drawing". The *observer* can use general collaborative tools to ask the *creator* to "Pause Drawing" and to "Drop Control" so as to take the control and to edit the collaborated entity.

The fourth function "Manipulation" can be used for dynamic collaboration such as CAD geometry co-editing. An interpolated NURB spline can be co-edited by clicking "NURB curve" from the sub-function of 4.3, as shown in Figure 4.10. The fourth function can also be used to "Delete" and "Undelete" the collaborated entity. The "Insert TEXT" in the fifth function is used to create a text string to be used for "Engrave TEXT on a surface" from the sub-function 7.3.

The sixth function, as shown in Figure 4.9, provides several display functions including the wireframe, hidden line removal and shading on both single view or four views. The seventh function "Machining" allows the user to generate the tool path based on a free form surface, output the NC program and issue the command to actuate the remote machining cell. The eighth function "File Control" allows the user to save the collaborated results and the ninth function "Case studies" adopts some examples to illustrate the *COCADCAM* system, which will be explained and illustrated in Chapter 5.

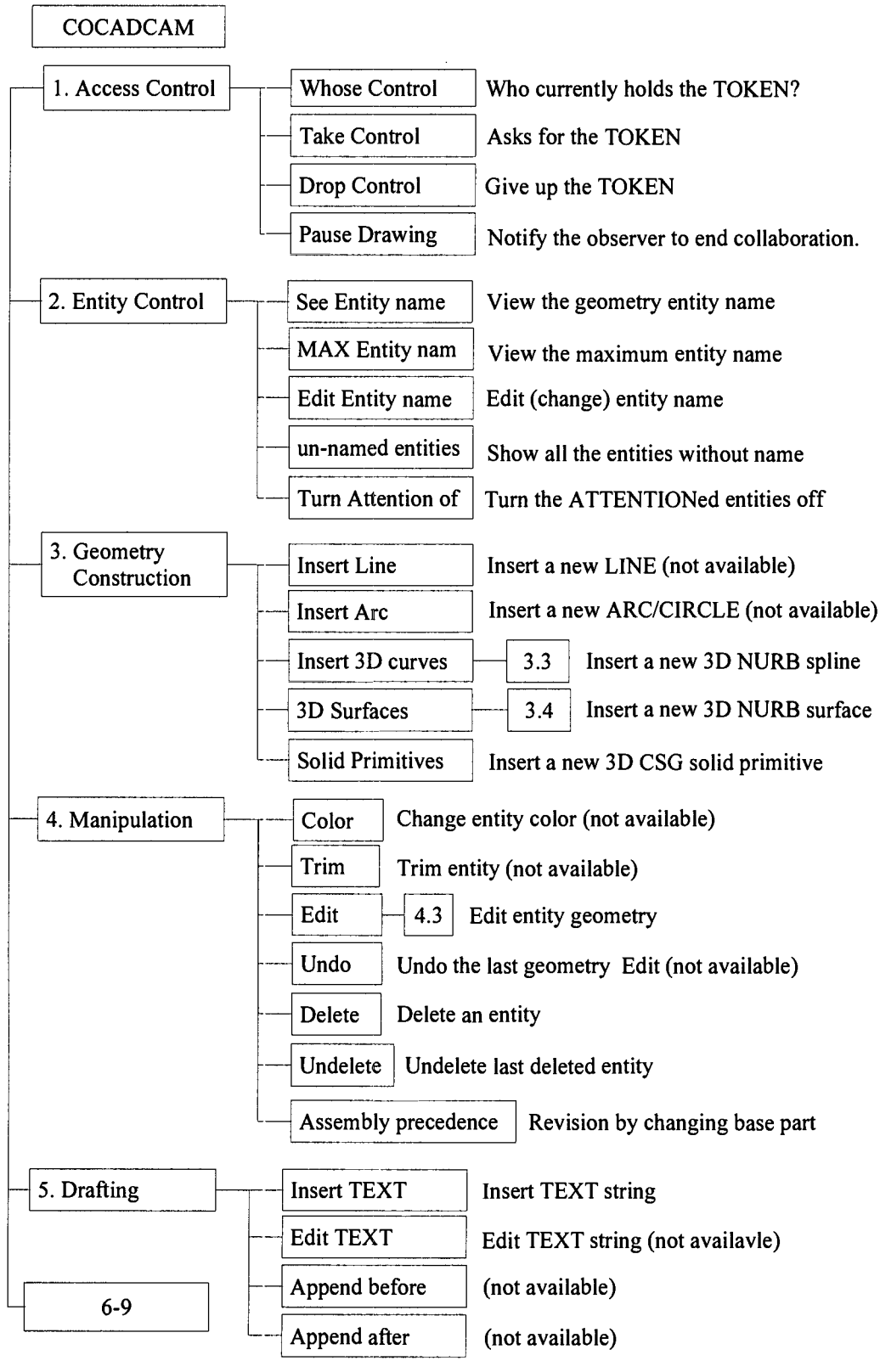


Figure 4.8 Program structure (functions 1-5)

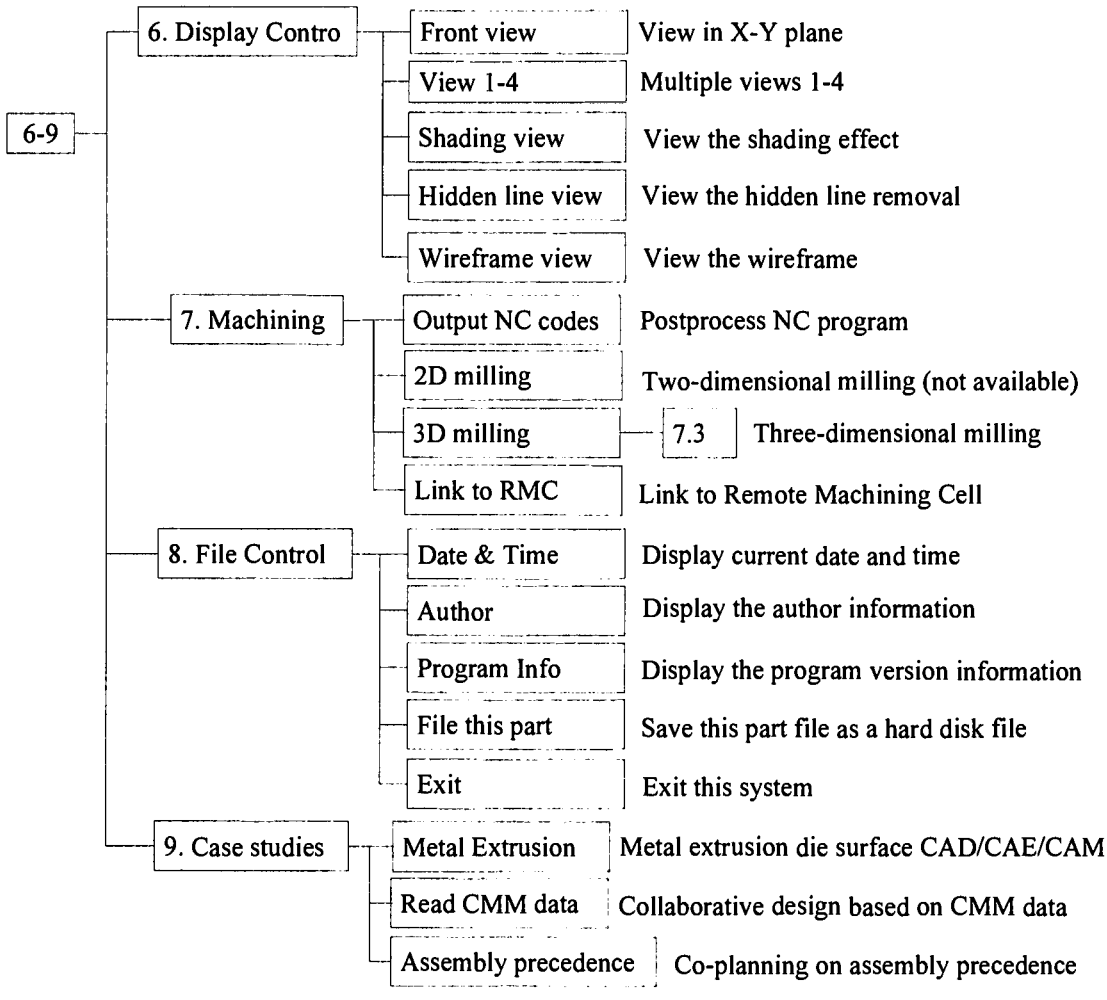


Figure 4.9 Program structure (functions 6-9)

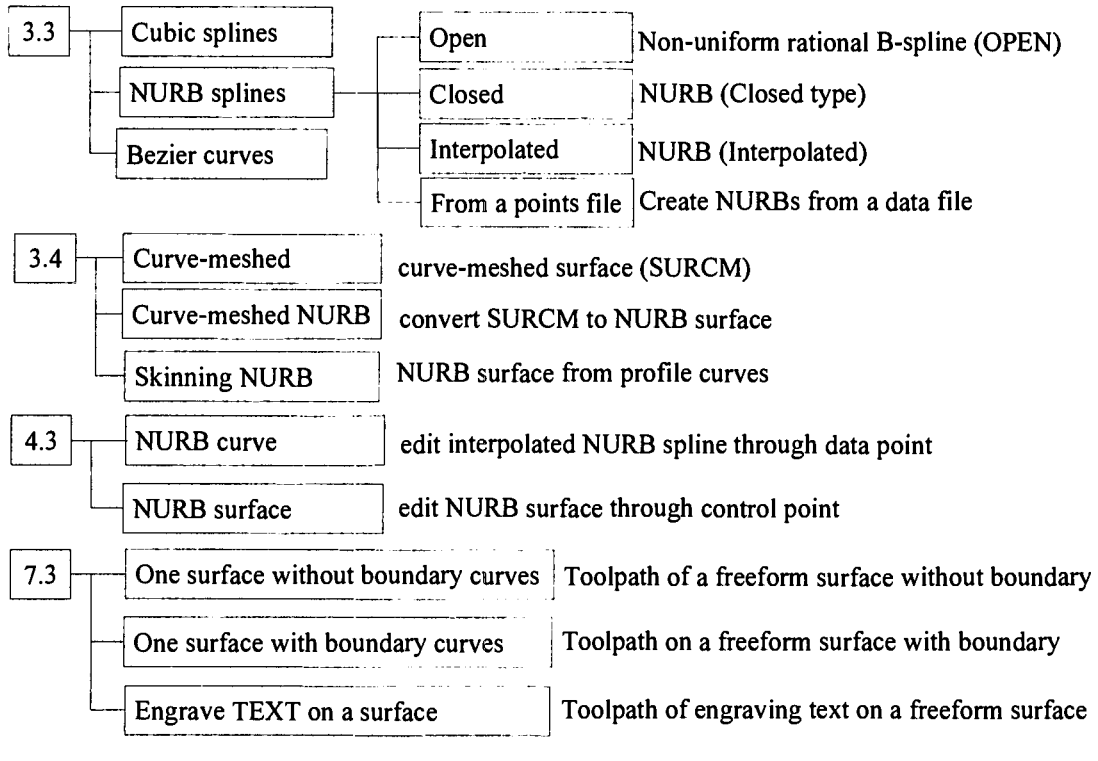


Figure 4.10 Program structure (sub-functions)

4.3 The COCAD Subsystem

4.3.1 Overview

The *COCAD* subsystem was designed to provide an environment allowing geographically dispersed team users to communicate and to dynamically revise CAD geometry and related tasks. So far, the subsystem has been used in LAN and the world-wide Internet. An ISDN link has also been implemented to form a dedicated communication channel avoiding the problem of the unreliable Internet. The implementation of ISDN has shown promising results and is also promising in terms of satisfying the potential users' desires for reliable connection. This is partly due to the data interchange formats proposed in this research to reduce the amount of data for network data transmission so as to shorten the network response time for fast interaction.

Network communication was the key to extend a single-location CAD/CAM to a multi-location application. The limitation of a single-location CAD/CAM was that it did not support the communication channel between the geographically separated users for necessary communication at the same time. Networking modules were therefore developed to support *COCAD* for CAD collaboration and data communication among related application programs. The application programs included server, client, and CAD/CAM software; the most popular communication and networking strategies of the TCP/IP protocol suite was adopted for network communication. The TCP/IP architecture (Santifellar 1991) accommodated the communication technology in its subnetwork level, such as ARPANET, Satellite network, X.25, Ethernet, ISDN, and Serial Line Internet Protocol (SLIP). The utilisation of TCP/IP architecture was transparent to developers and users at the application level (Comer 1991).

To realise the *COCAD* subsystem, Sections 4.3.2, 4.3.3, 4.3.4 and 4.3.5 explain the system structure and the linkage to a traditional CAD/CAM system. These Sections consist of the followings:

1) the structure of the *COCAD* subsystem configuration and communication,
2) the relations between *COCAD* and a CAD/CAM system,
3) the data communication algorithm, and
4) the data interchange format for *COCAD* communication.

4.3.2 The Structure of the *COCAD* Subsystem Configuration and Communication

The system structure of *COCAD* depended upon the infrastructure of the adopted CAD/CAM system. The development of *COCAD* was based on the application programming interface (API) provided by the CAD/CAM system in order to access and manipulate CAD-geometry data. Generally, the system structure of *COCAD* can have two types, as shown in Figure 4.11. The first type is to construct *COCAD* totally based on the API of a CAD/CAM system if the API provides the user with the networking libraries. The other is to develop external networking modules (external programs) for network communication and to integrate them with the API functions. The *COCAD* subsystem in this research is the second type due to the system architecture of the ANVIL-5000 CAD/CAM system adopted; the API of Anvil-5000 did not support socket connection.

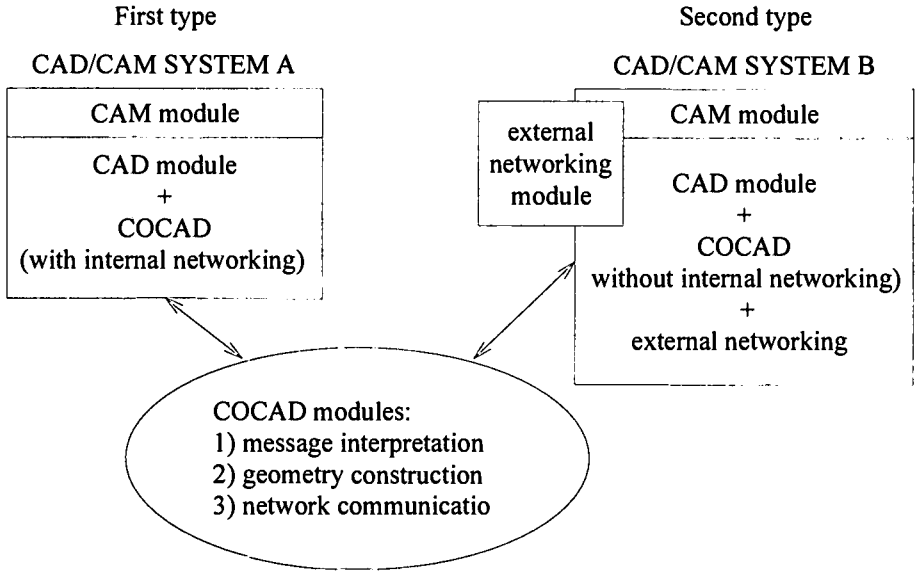


Figure 4.11 The structure of *COCAD*

The development of *COCAD* used the *XCALL* subroutine provided by the API of ANVIL-5000 which consisted of *GRAPL-IV* and *Extended GRAPL-IV*, linking to an external executable program (EP). The *XCALL* functions enabled an API program to exchange data with an EP and ran an EP which was external to the CAD/CAM system. This external program could be written in either FORTRAN or C (MCS 1993a). Both the API program and the EP were spawned through *XCALL*, ran in parallel, and the amount of data to be passed between these two processes was variable, so *'handshaking'* was needed. Data were exchanged between API and an EP, through a shared memory or similar

facilities. When an API program invoked an *XCALL* command, the data were passed and used as arguments. Constants were passed in one way to the EP while all the variables, of any length, were sent in entirety to the EP, and must be sent back in entirety. A maximum of twenty arguments was allowed for the data exchange at any one time. The communication procedures of the *XCALL* are shown in Figure 4.12 (MCS 1993a) based on the shared memory and semaphores of interprocess communication. The shared memory buffer (SMB) was used for the data exchange and the data movement is shown in Figure 4.13 (Stevens 1994). Shared memory was an algorithm for two or more processes to share a memory buffer. (Sharing a common piece of memory is similar to sharing a disk file.) A process means a program that is running under a certain operating system. The role of the external programs, as shown in Figure 4.13, controls the socket communication which links the CAD/CAM system to the other computer platform through a network. The role of the API (or *GRAPL-IV*) program and the external program is just like the client-server in a standalone computer. Semaphores were used for synchronisation. A sample time line for sharing single memory buffer between two processes, for example, *GRAPL-IV* and an external program, doing read-write data sharing through a shared memory buffer is shown in Figure 4.14 (Stevens 1994).

The general steps to share a memory buffer for data exchange is listed as follows.

(i) The server obtains control of a shared memory buffer using a semaphore.
(ii) The server places the data into the shared memory buffer.
(iii) The server notifies the client via semaphore to access the shared memory buffer and waits for the client's operation.
(iv) The client receives the data from the shared memory buffer, does the necessary manipulation, and notifies the server to access the shared memory buffer through the semaphore again.
(v) Steps (ii) ~ (iv) are repeated until finished.

The algorithm of interprocess communication was not supported by every CAD/CAM system on every computer platform. Therefore, an equivalent function to *XCALL* could be to exchange data through data files if the API of the CAD/CAM system could make a system call to execute an external program.

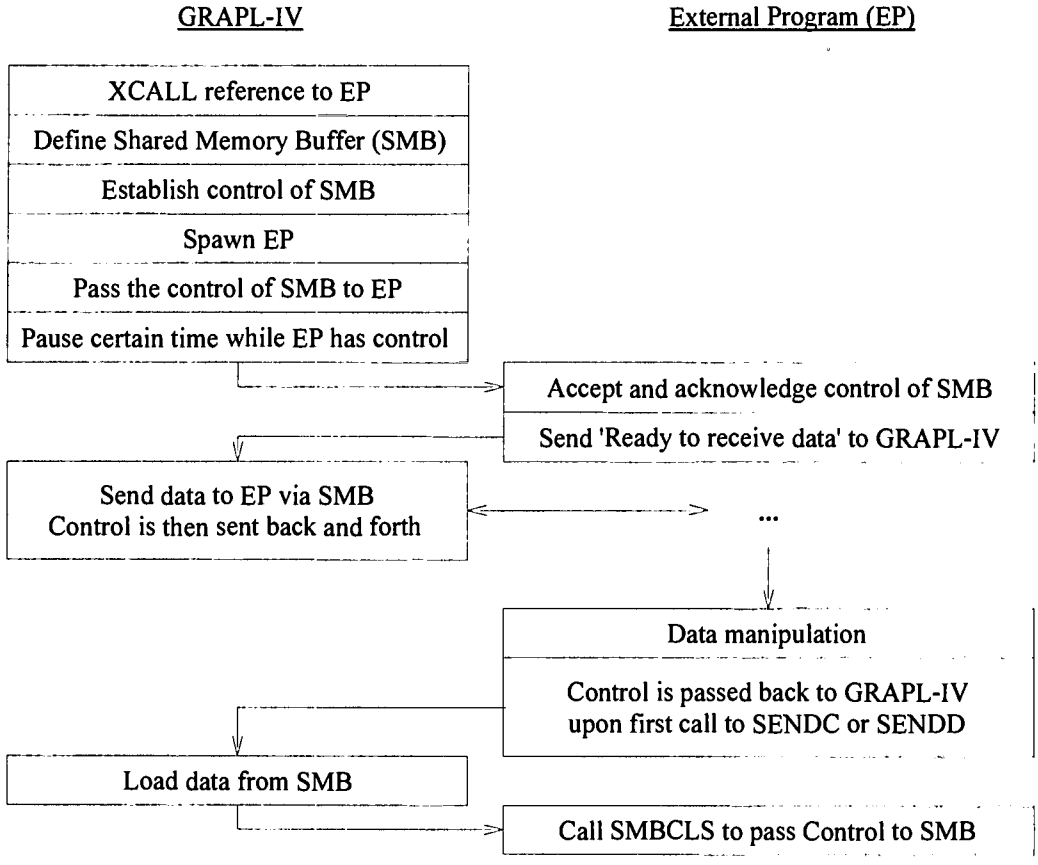


Figure 4.12 The outline of UNIX *XCALL* system communication

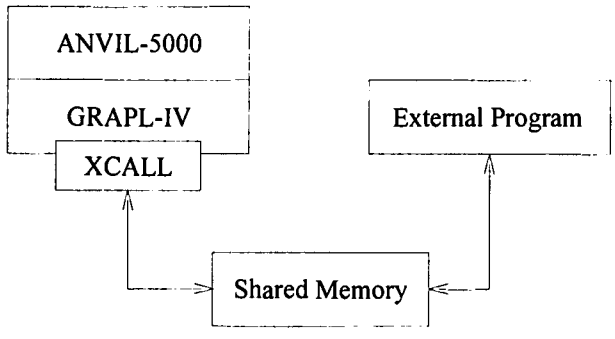


Figure 4.13 Movement of data between *GRAPL-IV* and external program (EP)

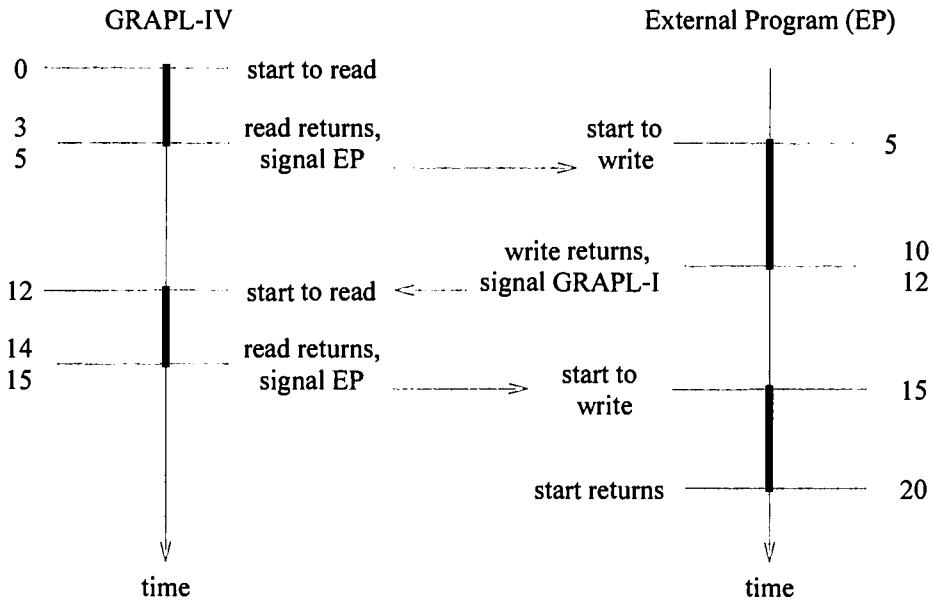


Figure 4.14 Example of time lines for read-write data between GRAPL-IV and EP

4.3.3 The Relation between *COCAD* and a CAD/CAM System

The development of *COCAD* was based on a typical CAD/CAM system. The purpose of this development was to explore and to study the requirements to enhance existing CAD/CAM systems for collaborative operation. Generally there were two alternatives in interfacing different CAD/CAM systems. One was direct database conversion, as shown in Figure 4.15, and the other was through neutral data interchange format, as shown in Figure 4.16. Although neutral file data exchange, for example, Initial Graphics Exchange Specifications (IGES) and Data eXchange File (DXF), had been popularly adopted, there were many dedicated modules developed for direct data exchange. This was because neutral file contained only general CAD data. There was no guarantee of 100 percent data exchange, even when the exchange was for the same CAD/CAM software to output and input the same data. Both of these provide a channel for the data exchange *as a whole* or on *selected* CAD geometry. If there was only one curve geometry out of thousands to be edited, either the whole CAD data or the complete curve data needed to be exchanged again. The other user must manually delete the original curve before or after the new curve data was received. This was resulted from the fact that usually the pre- and post-processor of data exchange did not automatically replace the original geometry with the new one. Therefore, special information was added to the data interchange format to enable *COCAD* to dynamically revise CAD geometry on the other side. The relationship between *COCAD* and the CAD/CAM system is shown in Figure 4.17. An external networking module was

used for the communication between CAD/CAM systems for synchronous CAD data revision based on the CAD data interchange formats, as will be presented in Section 4.3.5, proposed in *COCAD* for network communication. One of the important records in the data formats used by *COCAD* was the identification number (entity name), of the added geometric entity for each geometry to be traced for collaborative co-editing, if it was necessary.

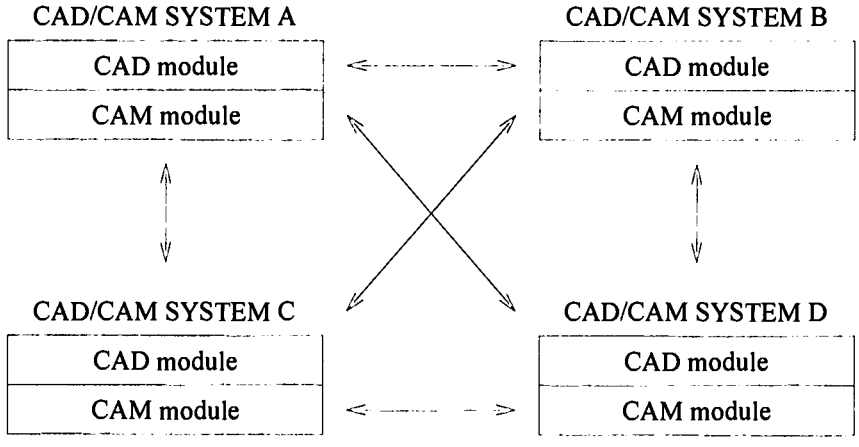


Figure 4.15 The direct CAD/CAM interface process

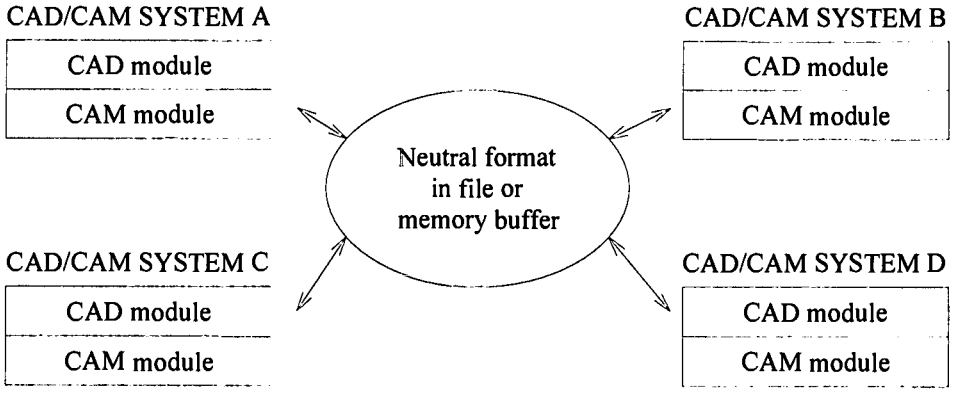


Figure 4.16 CAD/CAM interface through neutral files or memory buffer

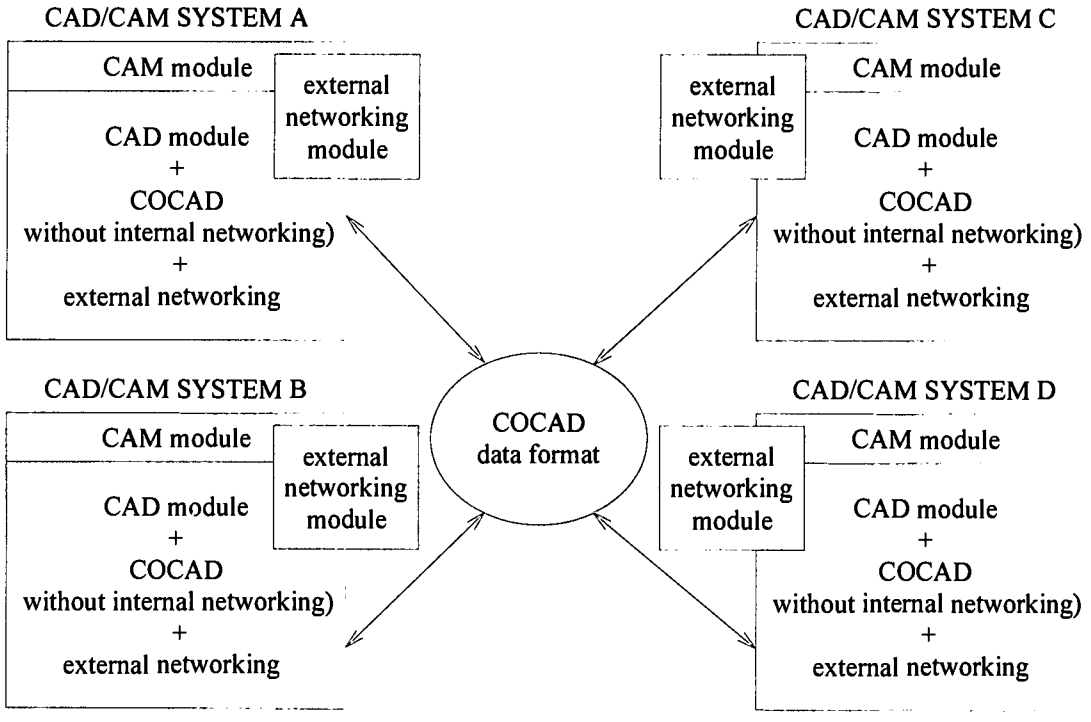


Figure 4.17 The role of *COCAD* data format among CAD/CAMs

4.3.4 Data Communication Algorithm

The data communication between the local server *camrs* and the local client *camrl* linking *COCAD*, as shown in Figure 4.2, was based on the communication strategy of a connection-oriented socket-based interprocess communication and several local disk files. The local server *camrs* and the local client *camrl* should be executed on the same computer platform. '*camrl*' was to be actuated by *COCAD* through the external call of the API to link with *camrs*. Examples of the external calls were *XCALL* in *GRAPL-IV* of Anvil-5000 or *XRUN* (*XSPAWN*) in *GRIP* of Unigraphics II (GRIP 1991). If the communication between *camrl* and *camrs* were on the *creator's* side, the communication flow would be in the form shown in Figures 4.18(a) and 4.18(b). On the *creator's* side, *camrl* contacted *camrs* through a predefined TCP port, as shown in Figure 4.18(a). '*camrl*' then stored the new data, that was output from *COCAD*, to a common disk file if *camrl* had the right to write to the common disk file. This right was obtained through checking a privilege file '*READLS.DAT*'. If the content of the privilege file was '1', the right belonged to *camrs*. If the content of the privilege file was '0', the right belongs to *camrl*. Once *camrs* was contacted by *camrl*, *camrs* forked itself to handle the communication with *camrl*. Once *camrl* finished writing the new data to the common disk file, *camrl* gave the privilege back

to *camrs* by changing the content of privilege file to '1'. *'camrs'* then contacted the global geometry server *ggserv* to update the data through a connection-oriented socket-based connection, as illustrated in Figure 4.18(b).

If the communication between *camrl* and *camrs* was on the *observer's* side the communication flow would look like Figures 4.19(a) and 4.19(b). On the *observer's* side, *camrl* contacted *camrs* through a predefined TCP port, as shown in Figure 4.19(a). Once *camrs* was contacted by *camrl*, *camrs* forked itself to handle the communication with *camrl*, and *camrl* stayed in a waiting status until *camrs* responded. *camrs* then contacted *ggserv* to obtain or to wait for the updated data through a connection-oriented socket connection, as illustrated in Figure 4.19(b). Once *camrs* received the updated data from *ggserv*, *camrs* checked the content of the privilege file *'READOB.DAT'* to see whether it had been offered the right to write the updated data to the common data file or not. If the content of the privilege file was '1', the right belonged to *camrs*. If the content of the privilege file was '0', the right belonged to *COCAD*. Once *camrs* had the privilege to write onto the common data file, it revised the common data file with the updated data from *ggserv* and returned access right to *camrl*. *'camrl'* was ended and the control went back to *COCAD* to manipulate the updated data if *camrl* had the response back from *camrs*. Once *COCAD* finished the manipulation of the updated data, *COCAD* returned the privilege of writing onto the common data file back to *camrs* by changing the content of the privilege file to '1'. Another client process *camrl* would be external-called to link to *camrs* for the next observation. The loop for the observation would not be ended until either the *creator* dropped the control or explicitly asked the *observer* to pause the observation through other conversation tools.

The source codes for the communication are listed in Appendix C for reference. These codes are written in C language format developed in workstations running Solaris 2.4 operating system and ULTRIX 4.5 operating system.

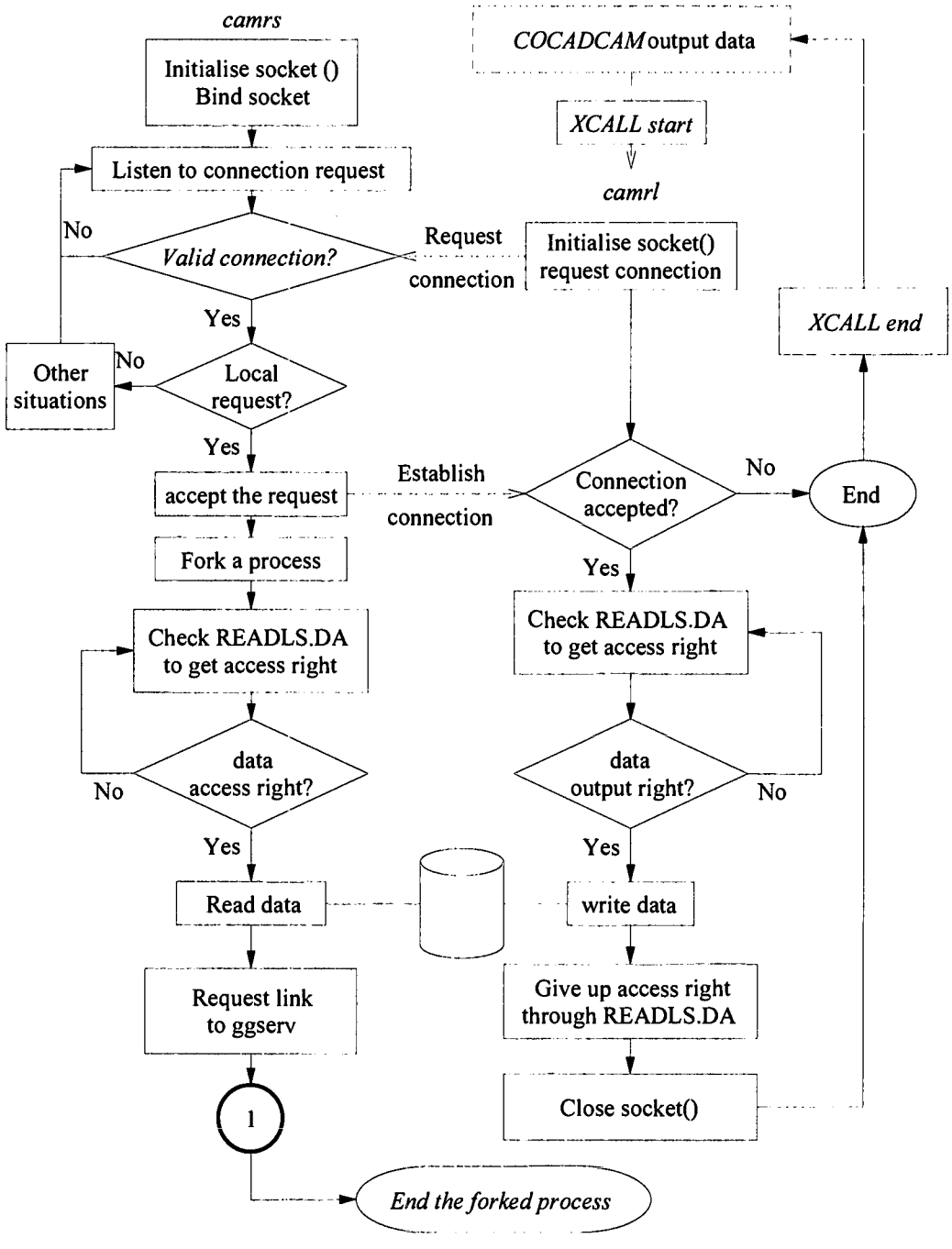


Figure 4.18(a) Communication between *camrl* and *camrs* on *creator's* side

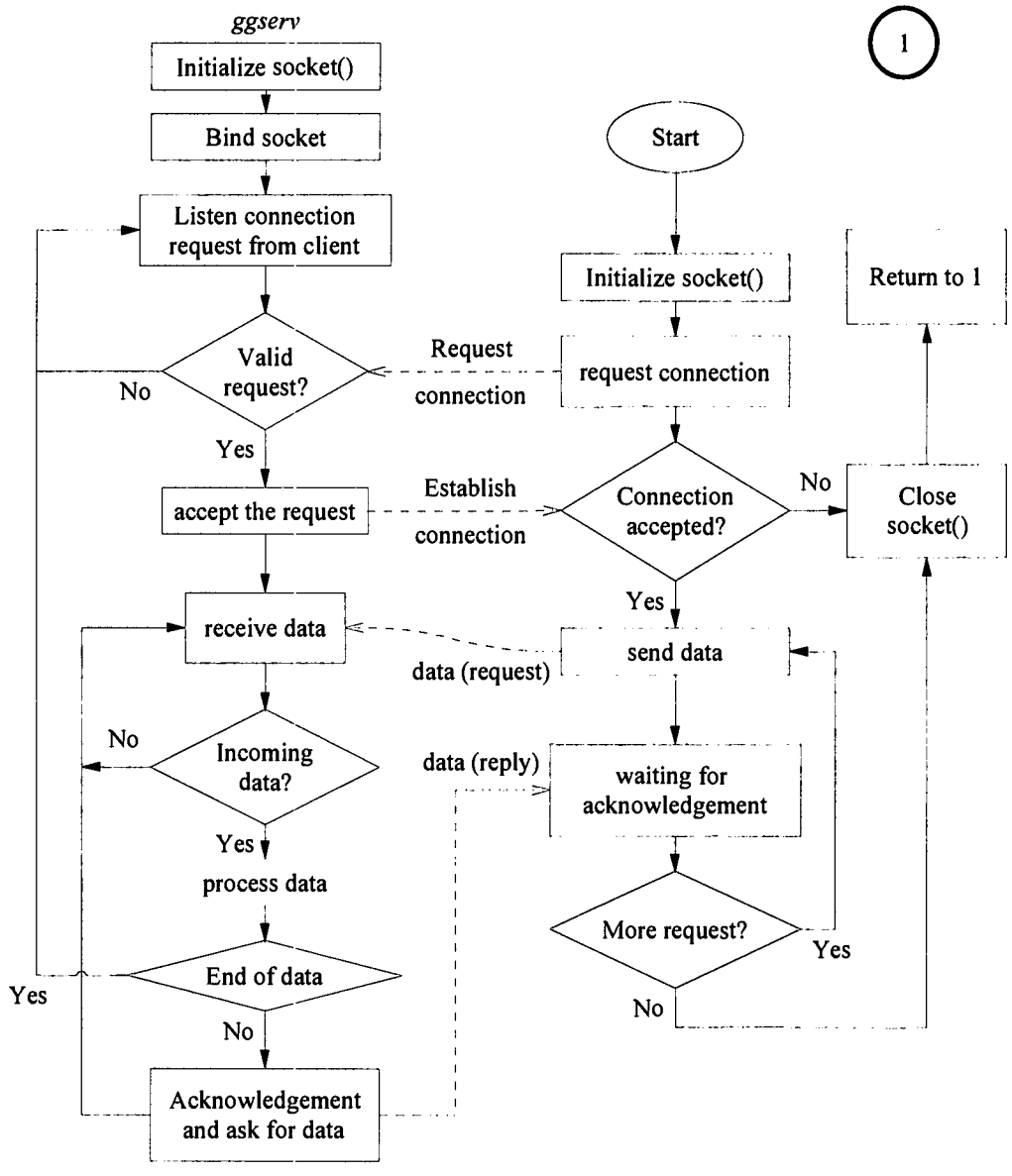


Figure 4.18(b) Communication between *camrs* and *ggserv* on *creator's* side

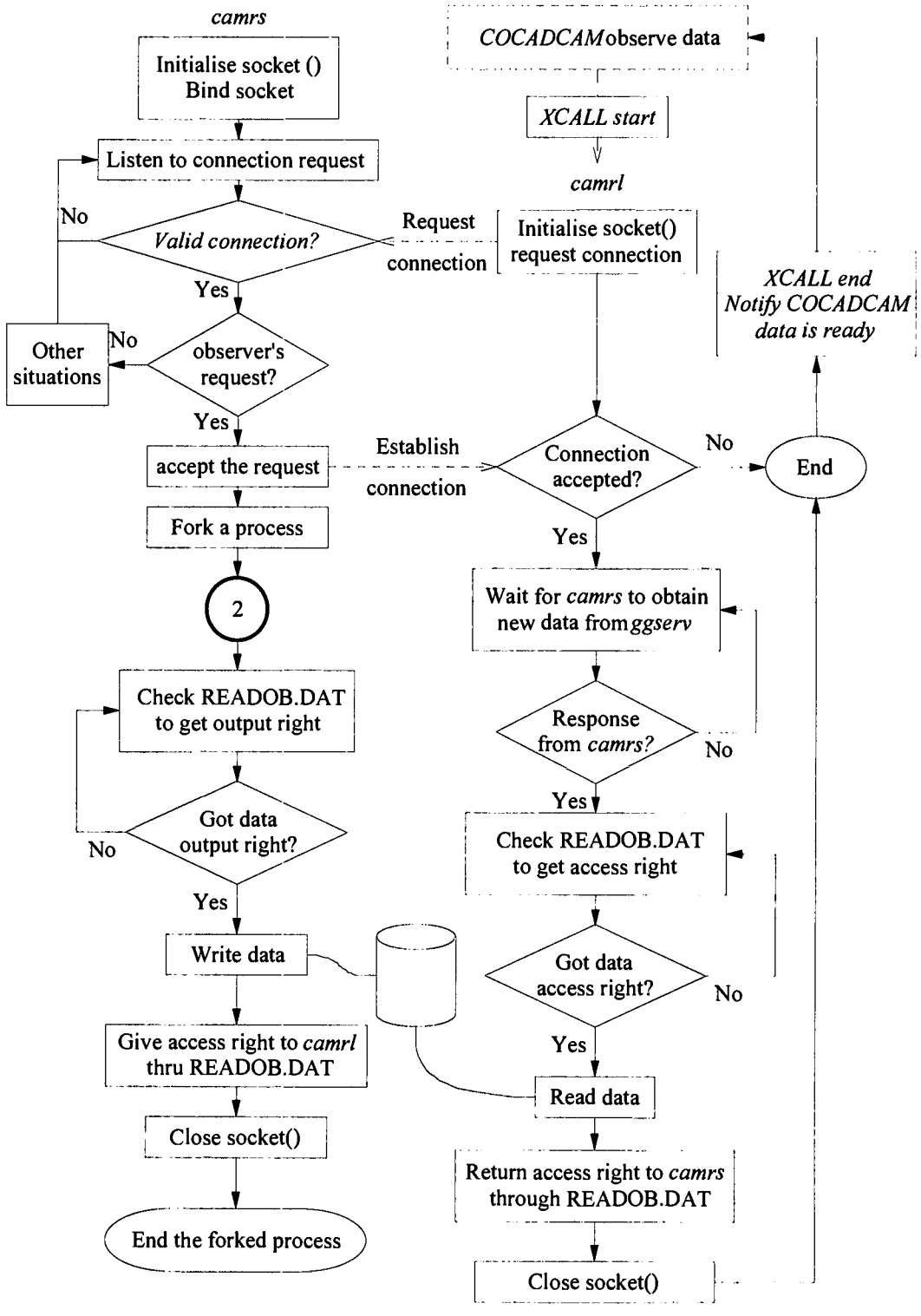


Figure 4.19(a) Communication between *camrl* and *camrs* on *observer's* side

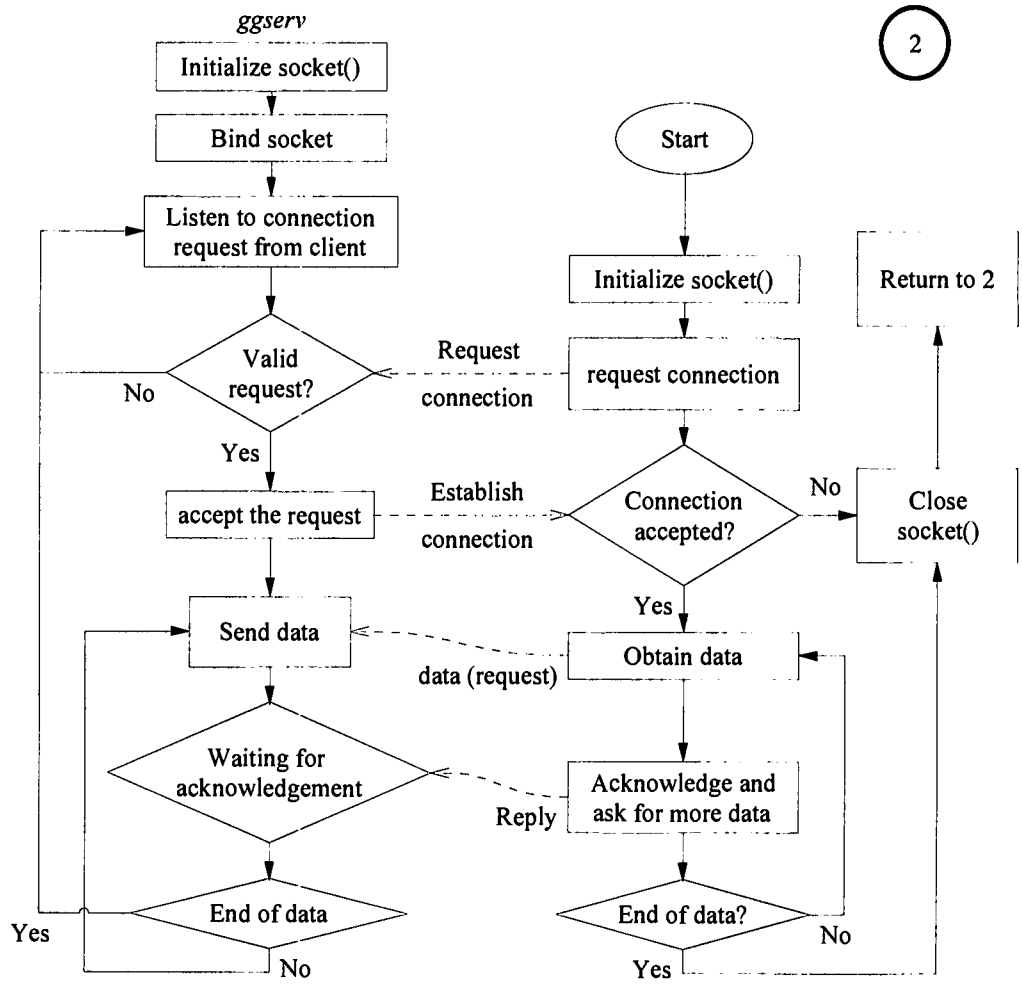


Figure 4.19(b) Communication between *ggserv* and *camrs* on *observer's* side

4.3.5 Data Interchange Format for *COCAD* Communication

This section explains the data interchange format for the communication in a collaborative CAD co-editing session. The idea behind the definition of the data format was to minimise the amount of data by allowing only those necessary for the transmission through network to shorten the network response time for dynamic and collaborative communication. This was based on the assumption that the major part of the computer calculation was handled locally, that is to say the CAD/CAM workstation was powerful enough for efficient calculation. A sample data format for the communication of the CAD geometry is shown in Table 4.1. The first record in Table 4.1 is the *'Owner ID'* where the IP address is used to distinguish the user who creates the geometric entity from the other. The second record is the *'Command Name'* which could be EDIT, INSERT, DELETE, UNDELETE, TRIM or other keywords. The third record is the *'Geometry Type'* definition. The *'Geometry Type'*

could be NURBI, for interpolated non-uniform rational B-spline (NURB) curve, NURBC, for closed NURB curve, TP3, for three-axis toolpath, or SURCM, for curve-meshed surface, etc. The fourth record is the *'Geometry Name'* (entity name) which is unique to a specific geometry. The *'Geometry Name'* was designed to keep track of the geometric entity. For example, Anvil-5000 used sequence number and other information to keep track of geometric entity but the sequence number assigned by Anvil-5000 could not be changed by the user. The records from *'Data records start'* to *'Data records end'* stored the data that were required by *'Command Name'* and *'Geometry Type'*. For example, if the *'Command Name'* was *'INSERT'* and the *'Geometry Type'* was *'NURBI'*, the fifth record in the table would be the total number of points. These data points defined the interpolated NURB curve followed by the Cartesian coordinates *'X Y Z'* of the points set. If the *'Command Name'* was *'EDIT'* and the *'Geometry Type'* was *'NURBI'*, the fifth record would be the *'Geometry Name'* of the point; and the new Cartesian coordinates of the point were followed to edit the interpolated NURB curve.

Table 4.1 The communication data format definition

Owner ID (IP address)	xx.xx.xx.xx	xx.xx.xx.xx	xx.xx.xx.xx
Command Name	INSERT	EDIT	INSERT
Geometry Type	NURBI	NURBI	SURCM
Geometry Name	12	30	40
Data records start	40	5	No. of U curves
...	X1 Y1 Z1	X5 Y5 Z5	...
...	...	NULL	No. of V curves
Data records end	X40 Y40 Z40	NULL	...

Three sample communication data are shown in Table 4.2. Sample 1 shows a data message from the user that is on a host computer with IP address 130.220.19.77. This message asks for an insertion (INSERT) of an interpolated non-uniform rational B-spline (NURBI) which has the entity name '10' and consists of five points. These points are (10.0 10.0 10.0), (20.0 20.0 25.0), (30.0 30.0 30.0), (40.0 40.0 20.0) and (50.0 50.0 10.0). Sample 2 is a message from host 130.220.19.76 which asks for editing (EDIT) an interpolated NURB spline that has the entity name '10'. The second data point of the interpolated NURB spline '10' is to be changed to (20.0 20.0 10.0). Sample 3 is a message from host 130.220.19.76 and asks for insertion of a curve-meshed surface (SURCM) which has the entity name '40'. This curve-meshed surface consists of three fixed (U) curves ('10', '11' and '16') and two variable (V) curves ('12' and '20'). These messages are sent to the

other designer sequentially, based on the connection-oriented client and server model, during the collaborative co-editing session. The source codes on the data interpretation, geometry construction and manipulation in the *COCAD* subsystem are listed in Appendix D for reference.

Table 4.2 The sample communication data

Sample 1	Sample 2	Sample 3
130.220.19.77	130.220.19.76	130.220.19.76
INSERT	EDIT	INSERT
NURBI	NURBI	SURCM
10	10	40
5	2	3
10.0 10.0 10.0	20.0 20.0 10.0	10
20.0 20.0 25.0		11
30.0 30.0 30.0		16
40.0 40.0 20.0		2
50.0 50.0 10.0		12
		20

4.3.6 Realisation of *COCAD*

This section uses an example to illustrate *COCAD* and presents an application in co-editing non-uniform rational B-spline (NURB) geometry based on a traditional single-location CAD/CAM software. This example shows how the *COCAD* subsystem facilitates the designers at geographically dispersed locations working synchronously as a team. Distributed designers could co-edit the interpolated NURB splines and a NURB surface that is based on a curve-meshed surface consisting of interpolated NURB splines geometry data at a distance, through the *COCAD* subsystem.

The graphical user interface environment of *COCAD* is based on a CAD/CAM system, as shown in Figure 4.20, where there are five interpolated NURB splines: u1, u2, u3, and v1, v2. To start the example, assume a curve-meshed free-form surface has been constructed and both geographically dispersed designers A and B have a copy of these NURB splines. The example can also be started from the beginning where there is no CAD geometry. Designer A can use e-mail to make an appointment for the collaborative work and then use text *'talk'* to contact Designer B to start the collaborative CAD co-editing session. Designer A can then ask Designer B to enable and to make sure all the necessary

collaborative tools such as those described in Table 2.1 are available. Once the collaborative tools are ready, the collaborative work is also ready to be started. Designer A can take control and create interpolated NURB splines u1, u2, u3, v1 and v2 one by one. The interpolated NURB splines are then transmitted to Designer B and the same geometry is shown on Designer B's screen. The collaborated results to realise *COCAD* are shown in Figures 4.21(a) and 4.21(b). The original NURB splines, as shown in Figure 4.21(a), are used to construct a curve-meshed surface, as shown in Figure 4.21(b). The u1, u2 and u3 curves are used as the fixed curves while v1 and v2 are used as the variable curves for constructing a curve-meshed surface. The co-edited NURB spline which is originally u2 in Figure 4.20 is shown in Figure 4.21(c) where the middle data point of u2 is co-edited (the Z coordinate is changed). An updated curve-meshed surface based on the co-edited u2 spline is shown in Figure 4.21(d). All figures from Figure 4.21(a) to Figure 4.21(d) are displayed on the screen of both *creator* and *observer* progressively during the collaboration. This example shows that *COCAD* can be used by two designers to co-edit NURB splines interactively and collaboratively to construct a free form surface. The free form surface can then be revised if one of its constructing three-dimensional entities is modified.

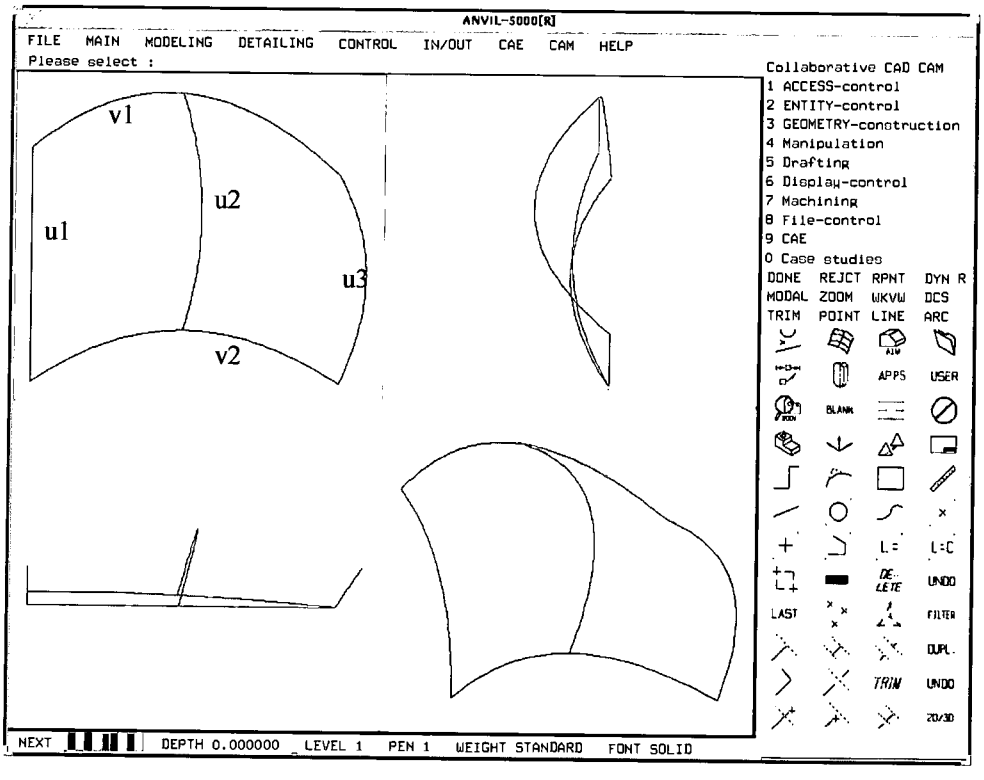


Figure 4.20 The graphical environment of *COCAD*

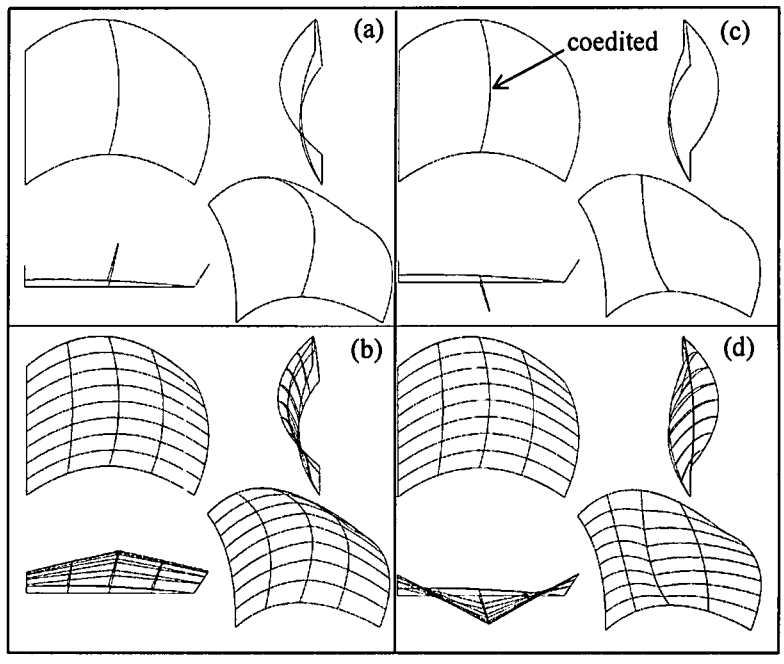


Figure 4.21 Collaborated results from *COCAD*

4.4 The COMAN Subsystem

4.4.1 Overview

This section describes the *COMAN* subsystem which controls the CAM function in the *COCADCAM* system. The *COCAD* subsystem, presented in Section 4.3, controls the network data interpretation, CAD entity construction, and manipulation. The *COMAN* subsystem is composed of three modules: toolpath generation, post-processing of the NC program, and linking a remote machining cell. The system structure of *COMAN* is shown in Figure 4.22. (The remote machining cell has been described in Chapter 3.)

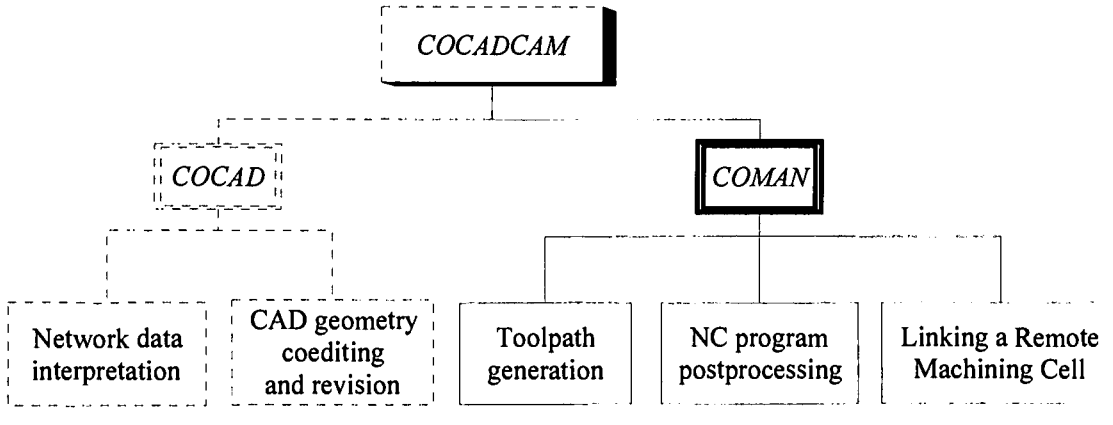


Figure 4.22 Components of *COMAN*

4.4.2 Toolpath Generation

The module for toolpath generation is based on the computer-aided manufacturing (CAM) statements of the API supported by a CAD/CAM system. Generally the computer-aided manufacturing statements support the subroutines necessary to create a milling toolpath. These statements include CAM modal settings, definition of cutting tools, management of tools' library, extraction of tool information, generation of customised postprocessor statements, assignment of the CAD geometry to be machined, and toolpath generation for NC program postprocessing. A general purpose NC postprocessor can be customised to generate the NC program for a specific CNC machine once the toolpath is created and its associated cutter location (CL) file is output in specified format. For example, the *NCPOST* was a general NC postprocessor in the adopted CAD/CAM system Anvil-5000 and the CL file was set to be in *CLPRNT format without surface points* for *NCPOST* to process. Since there was no single function to generate the toolpath of an existing geometry, to output a CL file, and to postprocess a NC program in either *GRAPL-IV* subroutine or Anvil-5000 system menu. Therefore, customised CAM functions were developed to link to a remote machining facility. So far, there have been three types of three-axis toolpath generation (TP3) developed: 1) MILLA - single free form surface without boundary geometry, 2) MILLB - single free form surface with boundary geometry and 3) MILLC - text engraving on a free form surface. The examples of communication data are shown in Table 4.3. The data format for the first type MILLA requires *'Command Name'*, *'Geometry Type'*, *'Geometry Name'*, *'Milling Type'*, *'Number of Surfaces'*, *'Name of Surface'*, and *'Tool Name'*. The data format for the second type MILLB requires boundary curves besides the data that are required in MILLA. The data format for MILLC requires a text string on top of the data for MILLA.

The toolpaths of these three types, as shown in Table 4.3, are illustrated in Figures 4.23, 4.24 and 4.25. The entity '201', as shown in Figure 4.23, represents a free form surface and entity '202' represents a toolpath that is of the type MILLA. The entities '6', '200', '4', '20', as shown in Figure 4.24, represent the boundary curves and the entity '203' represents a toolpath of the type MILLB. The entity '204', as shown in Figure 4.25, is a text string while '205' represents a toolpath that is of the type MILLC. The objective in developing these milling operations was to explore the feasibility in extending the collaborative CAD/CAM system with the capability of remote machining instead of developing a comprehensive milling toolpath generator. The process planning for the milling operations and the related parameters were assumed to have default settings. These

parameters included the spindle-speed, feed rates, workpiece and tool materials, etc. Therefore, once the toolpath was generated, the NC program of this toolpath could be generated immediately.

Table 4.3 The CAM communication data examples

Owner ID (IP address)	130.220.19.77	130.220.19.76	130.220.19.76
Command Name	INSERT	INSERT	INSERT
Geometry Type	TP3	TP3	TP3
Geometry Name	202	203	205
Milling Type	MILLA	MILLB	MILLC
Number of Surfaces	1	1	1
Name of Surface	201	201	201
Tool Name	m1	m1	m1
No. of Boundary curves or TEXT		4	1
Curve (TEXT) 1		6	204
Curve (TEXT) 2		200	
Curve (TEXT) 3		4	
Curve (TEXT) 4		20	
...		...	

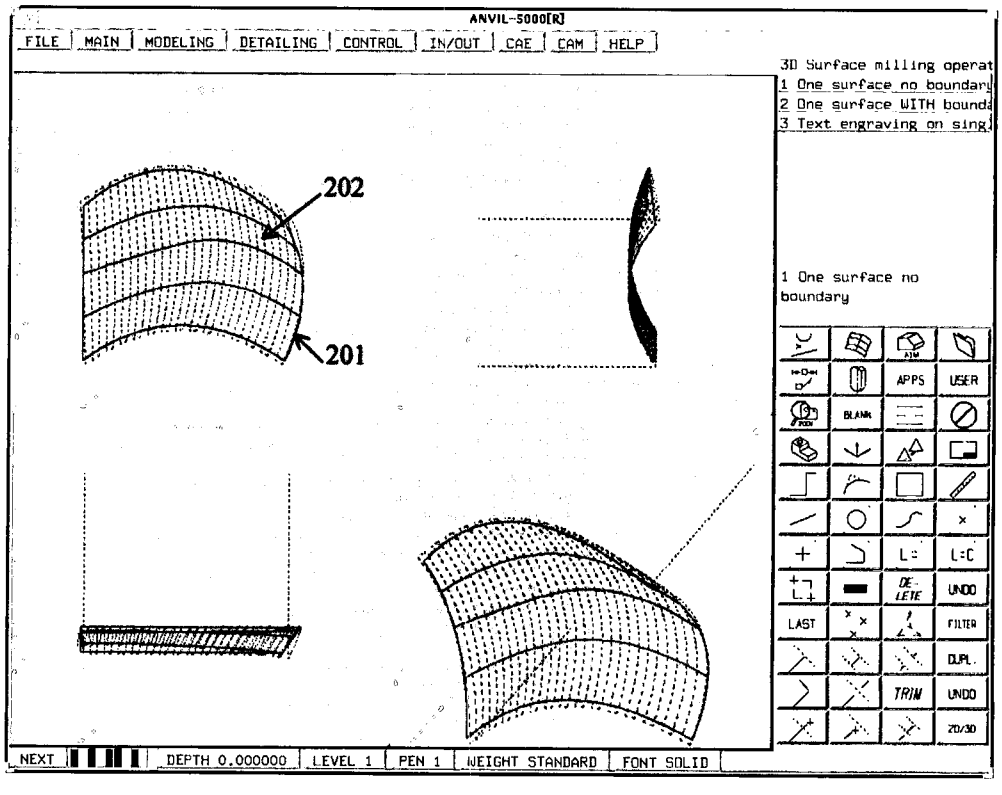


Figure 4.23 The toolpath of a single free form surface without boundary curves

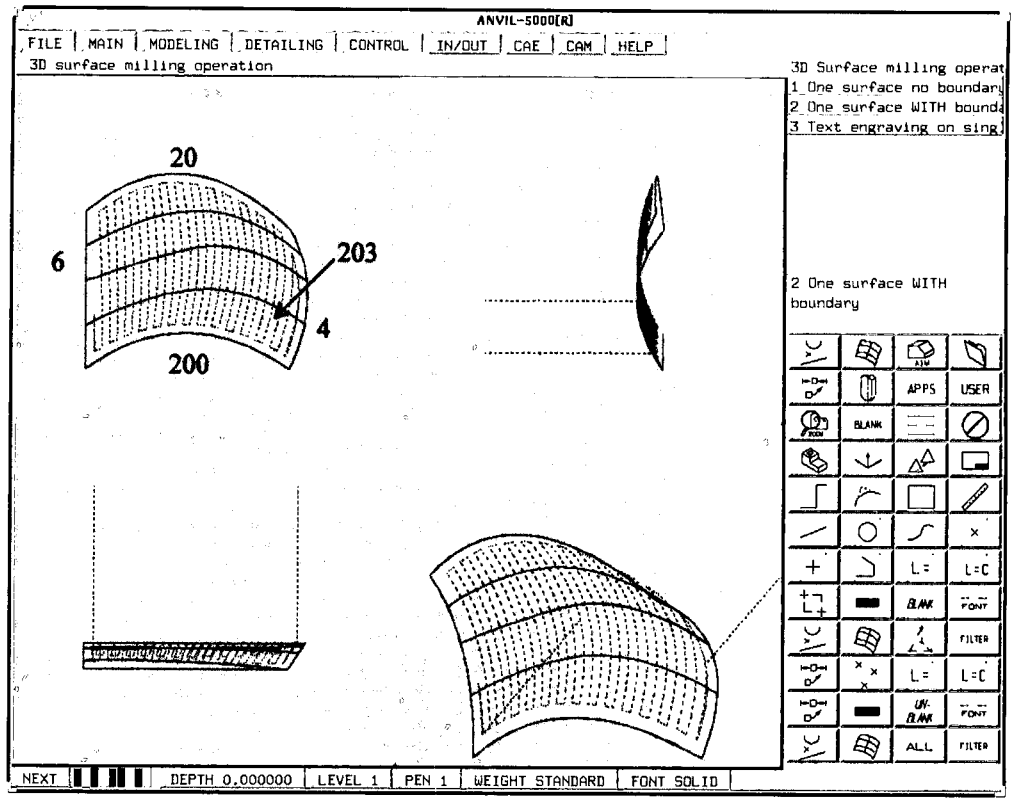


Figure 4.24 The toolpath of a single free form surface with boundary

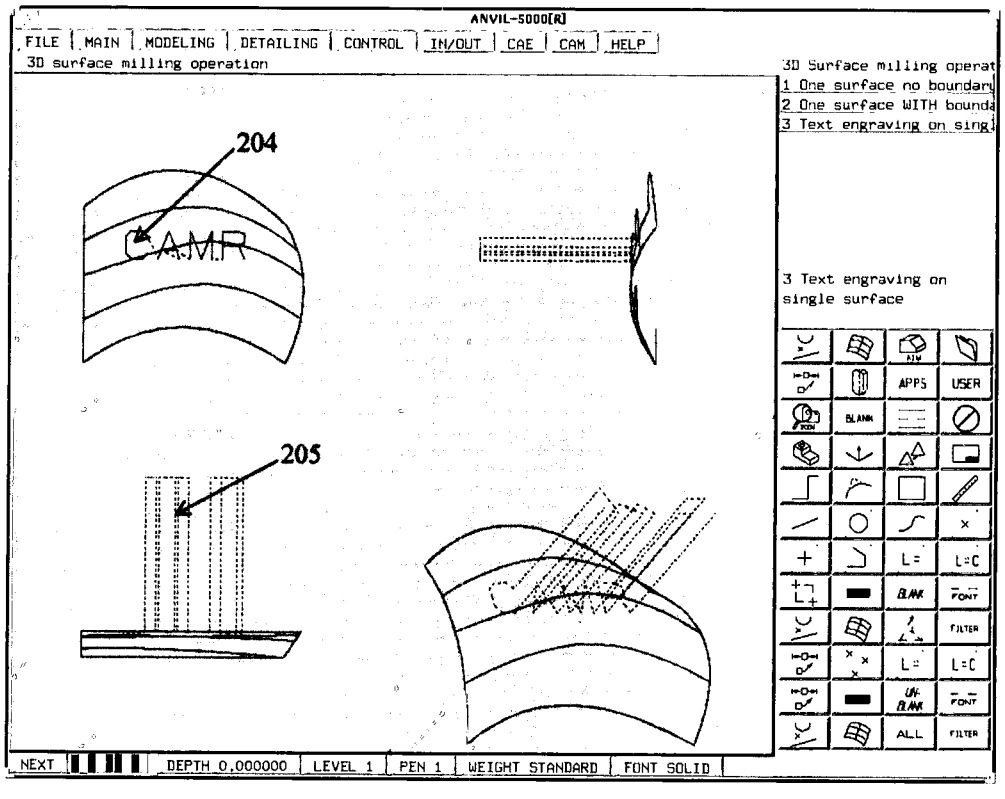


Figure 4.25 The toolpath of a text engraved on a single free form surface

4.4.3 Postprocessing of the NC Program

The post-processor for the NC program in the *COCAM* subsystem adopted the *NCPOST* function in Anvil-5000. *NCPOST* in Anvil-5000 version 4.0 was a *GRAPL-IV* executable, which could not be accessed from a user developed *GRAPL-IV*, and the postprocessing speed was very slow. Although the *NCPOST* in Anvil-5000 version 5.0 was much more efficient in the postprocessing speed and became a built-in function that could be selected from the interactive menu, the *NCPOST* was still not a *GRAPL-IV* linkable subroutine. Therefore, the combination of a *GRAPL-IV* program linked to a customised *keystroke* file was adopted and implemented. A sample keystroke file is shown in Table 4.4, where the CL file of toolpath '202' is postprocessed and a NC program named *'o1000.mcv'* is output.

Table 4.4 A sample keystroke file

```
5.0
2 $$ R.5.0 BY MCS              SYSTEM CONTROL              01 000010 1  000001
2 SYSTEM CONTROL               SELECTION CONTROL           05 091001 1  000005
2 SELECTION MODALS             SEL BY NAME/SEQ/CURSOR      02 091101 1  000002
4 SINGLE SELECT BY NAME/SEQUENCE NUMBER/SCREEN POSIT        Y 091103 1  000001
2 SELECTION MODALS                                         #F 091101 2  000001
2 $$ R.5.0 BY MCS              CAM                         10 000010 1  000010
2 CAM                          UTILITIES                   03 111412 1  000003
2 UTILITIES                    OUTPUT                      06 111568 1  000006
2 TOOL PATH OUTPUT             MODALS                      01 111575 1  000001
2 OUTPUT MODALS                GLOBAL CONTROL              02 111303 1  000002
2 GLOBAL OUTPUT CONTROL        ALL MODALS TO DEFAULT       01 111430 1  000001
4 SET MODALS TO SYSTEM DEFAULT?                             Y 111501 1  000001
2 GLOBAL OUTPUT CONTROL                                    OC 111430 2  000010
2 OUTPUT MODALS                OUTPUT FORMAT               03 111303 1  000003
2 OUTPUT FORMAT                CLPRNT (USE FOR NCPOST)     02 111301 1  000002
4 SET CIRCLE RECORD TYPE TO SURFACE WITHOUT POINTS?         Y 116053 1  000001
2 ASCII OUTPUT RECORD          VARIABLE                    02 111983 1  000002
2 OUTPUT FORMAT                                            OC 111301 2  000010
2 OUTPUT MODALS                OUTPUT FILE NAME            04 111303 1  000004
2 OUTPUT FILE NAME             ENTER NAME                  01 111293 1  000001
2 OUTPUT MODALS                PARTNO DATA                 05 111303 1  000005
2 PARTNO DATA                  ENTER TEXT                  01 111298 1  000001
2 OUTPUT MODALS                                            OC 111303 2  000010
2 TOOL PATH OUTPUT                                         OC 111575 2  000010
2 UTILITIES                                                OC 111568 2  000010
2 CAM                                                      #F 111412 2  000001
2 $$ R.5.0 BY MCS              CAM                         10 000010 1  000010
2 CAM                          UTILITIES                   03 111412 1  000003
2 UTILITIES                    OUTPUT                      06 111568 1  000006
2 TOOL PATH OUTPUT             SELECT TOOL PATHS           02 111575 1  000002
2 SELECTION METHOD             ENTITY NAME                 01 004054 1  000001
5 202                                                   03 OC 004057 2  000010
2 SELECTION METHOD                                         OC 004054 2  000010
5 O1000                                                 05 OC 111578 2  000010
5 o1000                                                 05 OC 097309 2  000010
4 FILE ALREADY EXISTS, OKAY TO OVERWRITE?                   Y 097301 1  000001
2 TOOL PATH OUTPUT             NCPOST THIS FILE            05 116051 1  000005
2 CHOOSE POSTPROCESSOR         MCV-1000CR (4-axisno        03 116004 1  000003
5 O1000                                                 05 OC 116045 2  000010
6 Press <DONE> or <REJECT> to acknowledge the messag       OC 116043 2  000010
2 UTILITIES                                                OC 111568 2  000010
2 CAM                                                      OC 111412 2  000010
2 $$ R.5.0 BY MCS                                          #F 000010 2  000001
2 $$ R.5.0 BY MCS              SYSTEM CONTROL              01 000010 1  000001
2 SYSTEM CONTROL               SELECTION CONTROL           05 091001 1  000005
2 SELECTION MODALS             SELECT BY CURSOR            01 091101 1  000001
4 SINGLE SELECT BY SCREEN POSITION?                         Y 091102 1  000001
2 SELECTION MODALS                                         OC 091101 2  000010
2 SYSTEM CONTROL                                           OC 091001 2  000010
2 $$ R.5.0 BY MCS              SYSTEM CONTROL              01 000010 1  000001
2 SYSTEM CONTROL               KEYSTROKE MANAGEMENT        07 091001 1  000007
4 TERMINATE KEYSTROKE FILE CAPTURE?                         Y 106003 1  000001
```

4.4.4 Linking a Remote Machining Cell

The linkage to a remote machining cell was through the shared data files based on the network file system (NFS) between the UNIX system (Solaris 2.4 operating system) and the PC (Microsoft Windows for Workgroups Version 3.11). A text file was used as the command file serving as the communication bridge. This file was output from the *COMAN* subsystem. The content of this file is shown in Table 4.5 with nine data records. The left column of Table 4.5 is the command menu in the *COMAN* subsystem while the options are listed in the right column. The first record *'Which Machine'* is to select the milling machine where there is only one machining centre MCV-1000CR (*MCV*) available. The second record *'LOAD/UNLOAD'* is to select the method used for robot to handle the workpiece, where there are two options *VLOAD* (using vision system to assist loading tasks) and *UNLOAD*. The third record *'Workpiece'* is the encoded name of the workpiece, for example, *ALBLK* represents an aluminium block and *ALBAR* represents an aluminium bar. The workpiece can be located at either *BENCHA* or *BENCHB* (the fourth record). The fifth record *'clamping device'* is used to select the device to clamp the workpiece which can be either *VICE* or *CHUCK*. The sixth record *'Neural'* is to decide the function that the robot is going to use to load or unload the workpiece. The seventh record *'Activity'* is to decide the activity that the remote machining cell needs to perform which can be *'download* the NC program' to the machining centre or 'check the *status*' of the machining centre. The eighth record *'NC Filename'* is to enter the filename of the NC program. The ninth record *'Operation'* is to select the operation that the machining centre needs to do. The content of this file is assigned by the user interactively. The interactive menu for the initial command options is shown at the upper-right corner of Figure 4.26 and a sample command file content is shown at the upper-right corner of Figure 4.27.

Table 4.5 The command file linking a remote machining cell

Command menu	Options
Which Machine	*MCV*
LOAD/UNLOAD	*VLOAD, UNLOAD*
Workpiece	*ALBLK, ALBAR*
From Table	*BENCH A, BENCH B*
Clamping device	*VICE, CHUCK*
Neural	*PXPYC, LDC, DOWNZ*
Activity	*DOWNLOAD, STATUS*
NC Filename	*o1000.mcv*
Operation	*CUT, REPORT*

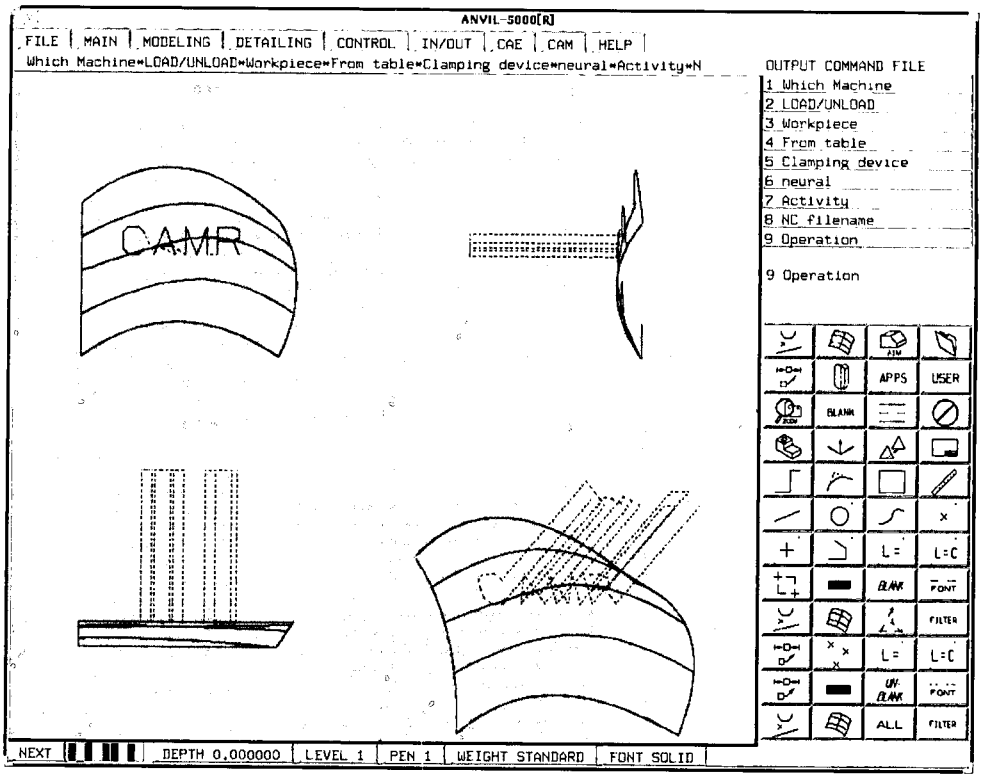


Figure 4.26 The command file menu linking remote machining facility

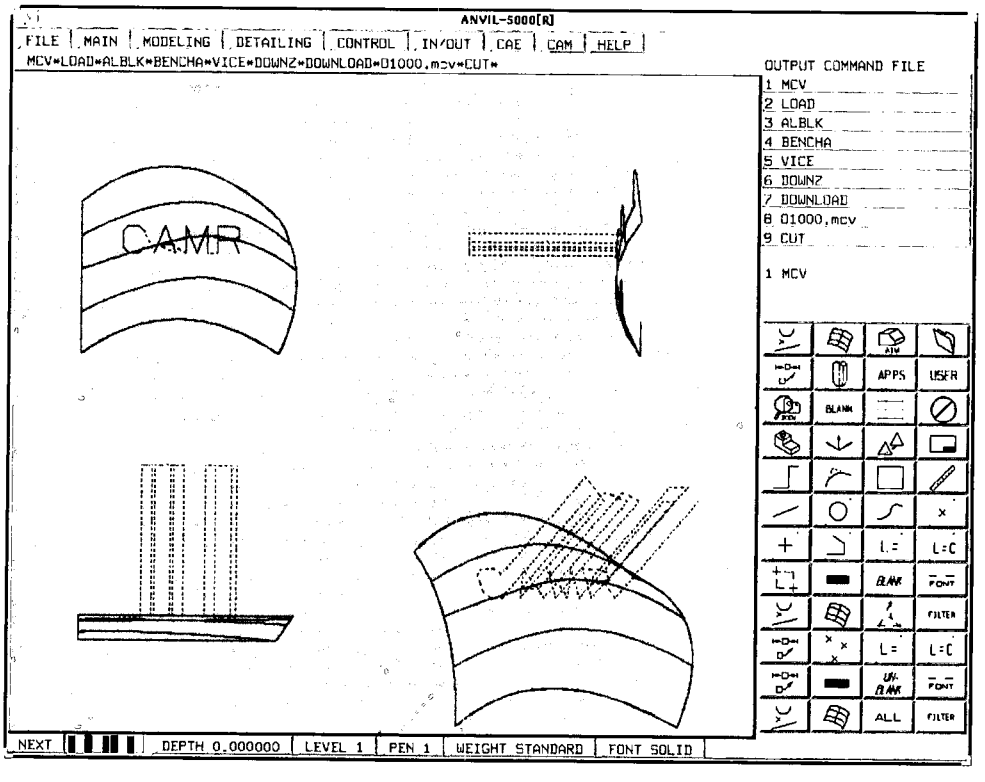


Figure 4.27 The command file contents linking remote machining facility

4.5 Concluding Remarks

A collaborative CAD/CAM system, *COCADCAM*, has been implemented and documented in this chapter. *COCADCAM* consists of two major subsystems: 1) a collaborative CAD subsystem, *COCAD*, and 2) a collaborative CAM subsystem, *COMAN*. The algorithm adopted is presented and described in this chapter to construct an environment for CAD collaborative work and the associated milling operation remotely. The *COCADCAM* system has been realised in a local area network and the Internet on top of the TCP/IP protocol suite. The Integrated Services Digital Network (ISDN) was also implemented forming a dedicated data communication channel to resolve the unreliable Internet bandwidth. The data format used in this development was designed to reduce the amount of the data in network transmission by transferring only the necessary data. With the connection-oriented client and server model for network communication, *COCADCAM* has become a synchronous system that can be used for collaborative work by geographically dispersed CAD/CAM users. The application of the *COCADCAM* system and the linkage to the remote machining cell is presented in Chapter 5.

The system structure of the *COCADCAM* system can be applied to other CAD/CAM systems. Although the development of *COCADCAM* described in this chapter adopted Anvil-5000 CAD/CAM software, the purpose was to implement the *COCADCAM* concept in using a system that supported both CAD and CAM functions through its application programming interface (API). Moreover, there were only two CAD/CAM systems available and accessible during the research for the implementation of *COCADCAM*: Anvil-5000 CAD/CAM system and Unigraphics II (UGII) CAD/CAM system. The CAD/CAM system UGII was purchased from McDonnell Douglas Corporation and was also successfully used to implement the *COCAD* subsystem of *COCADCAM* on a ULTRIX V4.5 operating system to validate the *COCADCAM* configuration. The GRaphics Interactive Programming Language (GRIP 1991) which was the API of UGII was used in the implementation. This meant a 3D curve created by a UGII user could be shown on the screen of a user of Anvil-5000 through the *COCADCAM* system. The source codes in implementing the *COCADCAM* system are listed in Appendix C and D.

Chapter 5 Illustrative Examples

5.1 Introductory Comments

This Chapter illustrates the application of the *COCADCAM* system by several examples: 1) the application of CAD-geometry co-editing in reverse engineering; 2) the distributed application in metal forming extrusion die design, optimisation, and machining; and 3) the collaborative design for the generation of a preliminary assembly sequence. The example on the reverse engineering application shows the *COCADCAM* system in the CAD design refinement for revising the CAD geometry by two geographically dispersed and networked CAD/CAM users (Kao and Lin 1996a, 1996b, 1997b). The next example shows a strategy to extend a traditional metal forming CAD/CAE/CAM integration to a distributed application: the design and analysis at one site, the generation of a tool path at the other site, and the remote machining at another site (Kao and Lin 1995b, 1996c, 1997a). The last example demonstrates an extended application of *COCADCAM* to a collaborative planning on preliminary assembly sequence generation (Lin and Hsu 1996, Hsu and Lin et al 1997) and modification between two geographically dispersed designers. The generation of a preliminary assembly sequence depended on the choice of a base part and available facilities, therefore, collaborative work was determined to be a feasible working style.

5.2 Application in Reverse Engineering

5.2.1 Overview

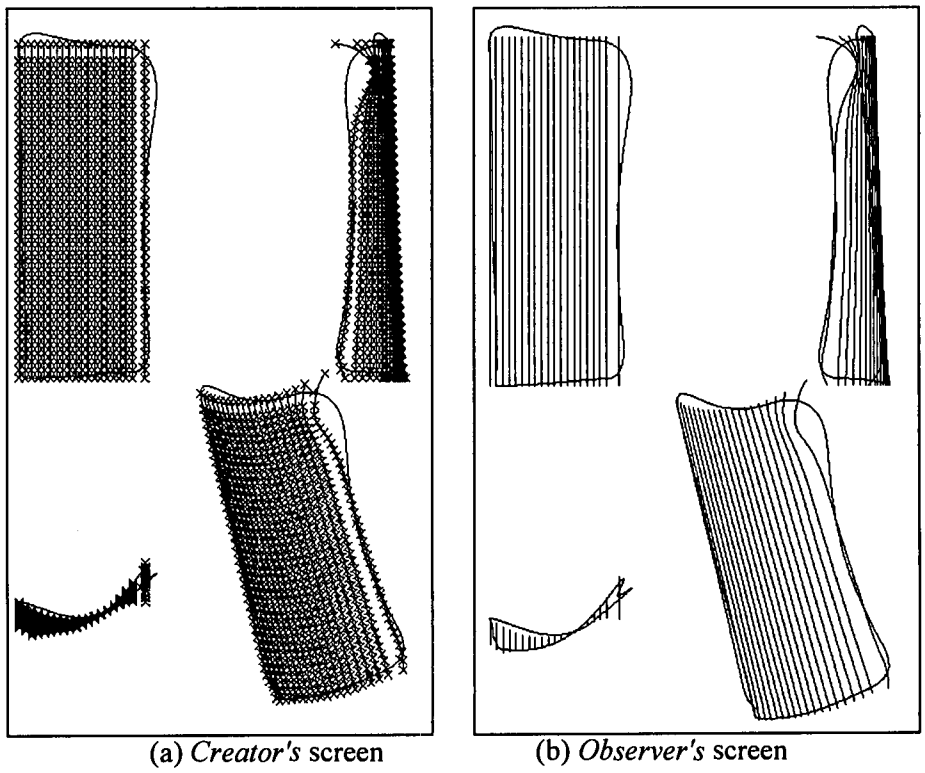
Two examples are used to illustrate the application of the *COCADCAM* system in collaborative CAD co-editing for reverse engineering tasks in this section. The first example illustrates the application of *COCADCAM* in collaboratively reverse engineering a turbine blade by using campus Ethernet local area network (LAN). The surface points of a turbine blade were assumed to have been measured by a coordinate measurement machine (CMM) and it was assumed that part of the measured data needed modification. To explore the *COCADCAM* system for international collaborative work between geographically separated engineers. The second example demonstrates a collaborative development of an automotive component, a horn pad, geometry reconstruction and revision through the

Internet and a dedicated ISDN connection. It was assumed that the central part of the horn pad geometry needed revision.

5.2.2 Reverse Engineering a Turbine Blade Collaboratively through LAN

A coordinate measurement machine (CMM) was adopted for the reverse engineering of a turbine blade surface. In general, the result of the CMM measurement needs either modification or further enhancement. The data points of a turbine blade surface, measured by a CMM, were used to construct a wireframe geometric model of the turbine blade. It was determined that only a little portion of the measurement needed modification, therefore, a collaborative session was conducted between the measurement engineer and the other CAD engineer.

To start this example, firstly, the data points of the turbine blade were assumed to have been measured by a CMM machine and were stored in a disk file by the measurement engineer. The *COCAD* subsystem of the *COCADCAM* system (at the measurement engineer's side) was used to read the data points, construct the geometry and transfer the points data to the global geometry server '*ggserv*' sequentially. These data points were then passed to the *COCAD* subsystem (at the CAD engineer's side) to construct the geometry. The data format in the network data transmission followed the definition in Table 4.1.



(a) *Creator's* screen (b) *Observer's* screen

Figure 5.1 The turbine blade geometry

Therefore, the interpolated non-uniform rational B-spline (NURB) curves were created and the wireframe model of the turbine blade geometry was then constructed on both sides, as shown in Figures 5.1(a) and 5.1(b). A snapshot of part of the *creator's* (measurement engineer's) screen showing both the data points and the interpolated NURB curves are illustrated in Figure 5.1(a). The NURB curves were constructed and displayed on the screen of the *observer* (CAD engineer), after received the data from the *ggserv*, as shown in Figure 5.1(b). The *creator* was the engineer who had the privilege to create and send the NURB splines geometry to *ggserv*. The *observer* was the other engineer who did not have the privilege to creating geometry but observing the operation of the *creator*.

The screen layout on the *creator's* computer is shown in Figure 5.2. The upper-right window shows the *COCAD* subsystem of *COCADCAM* in Anvil-5000, which also displays the wireframe geometry of a turbine blade. The lower-left window is the message window of the global server *gserv*. The lower-middle window shows the global geometry server *ggserv*. The lower-right window represents the local server *camrs*. The middle-left window shows a portion of the built-in text 'talk' tool in Sun workstation.

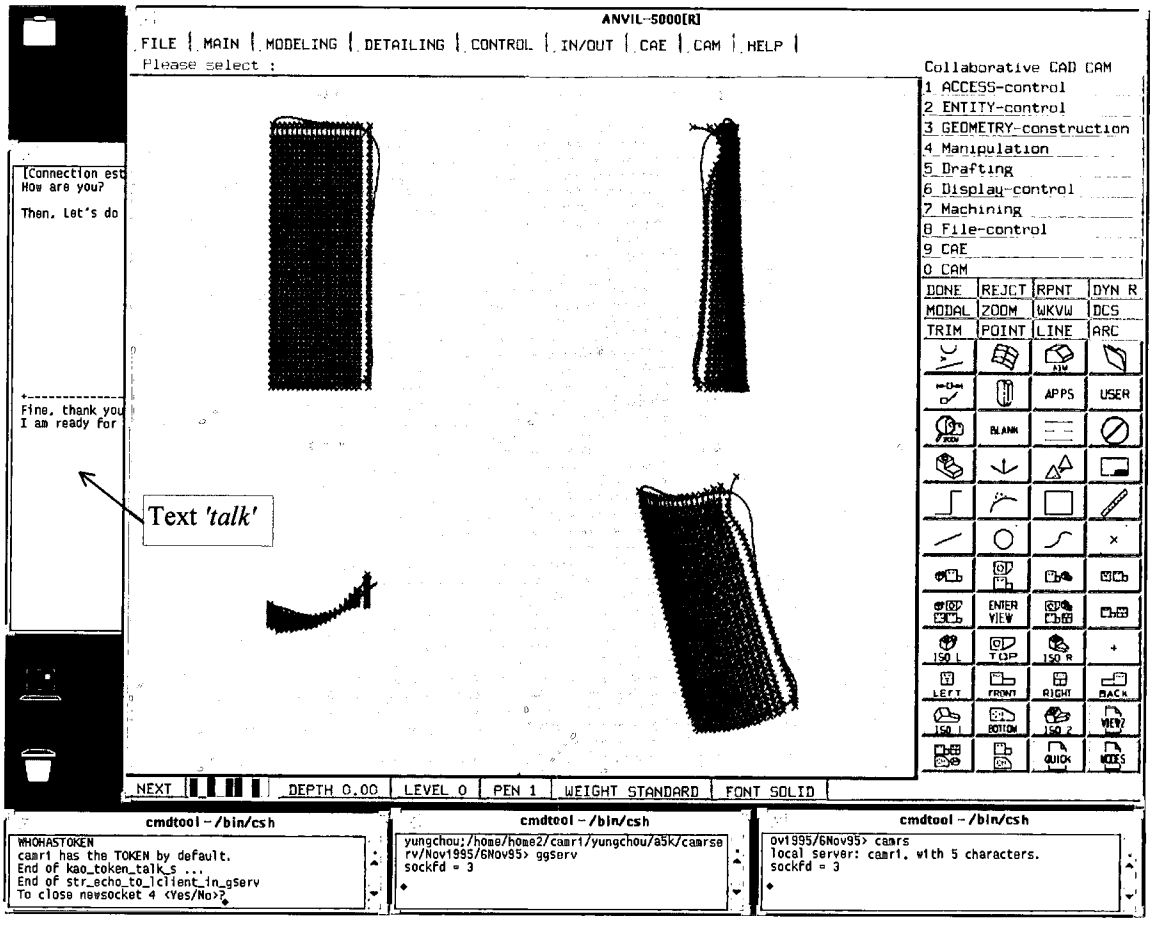


Figure 5.2 Screen layout of the *creator*

The conversation media between the measurement engineer and the CAD engineer were 'text', 'audio' and 'whiteboard'. The text 'talk' application program, in Sun workstation, was adopted for realtime text communication, such as to explicitly ask for the token, in the co-editing task of turbine blade. The command to execute the 'talk' program for text communication was 'talk username@hostname' where username was the login name and the hostname was the computer name or IP address. The telephone service and multi-media audio tools were also used for realtime oral communication. A Whiteboard tool was used as a discussion panel by attaching the picture, from the CAD/CAM application to the whiteboard window (Lin et al 1995a), and utilising the mark-up and annotation functions.

The data transmission route went through campus LAN where the transmission speed was 10 Mbits per second, and the computer platform was Sun 4/80 (SPARCstation 10). The Window system on Sun workstation was OpenWindows Version 3.4 and the operating system was SunOS release 5.4 generic (that is Solaris 2.4). The system layout is the same as that in Figure 4.7, where the computers were linked in a LAN.

The data base of Anvil-5000 CAD/CAM was accessed through *Extended GRAPL-IV* function in *COCAD*, and was used to facilitate the proposed data format, as shown in Tables 4.1 and 4.2. The data format was defined to accept, interpret, create, revise, and trace the geometric shape description in the communication. The co-editing process of the turbine blade is illustrated in Figure 5.3. After received and constructed the wireframe geometry of the turbine blade, the CAD engineer asked the measurement engineer for the privilege to revise part of the CAD geometry data where some suspicious situations were found. The CAD engineer then became the new *creator* and started the interactive and collaborative editing when the privilege was passed from the measurement engineer. The 'old point', as shown in Figure 5.3, of one of the NURB curves was then changed so that this NURB spline curve became an 'edited curve' with a different end point. The NURB curve geometry, as changed by the new *creator* (the CAD engineer), was reflected (displayed) on the new *observer's* (the measurement engineer's) screen. The reflection speed depended mostly on the network traffic, the software calculation time, and the time sharing mechanism of the computer operating system. These factors were out of a general user's control. However, the reflection of the geometry change happened very quickly, which took only a few seconds in this example. This fast reflection was expected because only the index of the edited NURB curve and the coordinates of the data point to be changed were transferred from the CAD engineer's side to the other. (The *COCADCAM*

system did not transfer the whole turbine geometry in co-editing session.) This example showed that *COCADCAM* could be applied in co-editing CAD geometry through LAN effectively.

A three-axis milling toolpath for the turbine blade surface, which was based on the collaboratively modified NURB curves, is shown in Figure 5.4. This toolpath could then be used by *COMAN* , the second subsystem of *COCADCAM*, to generate a NC program for milling machining operation.

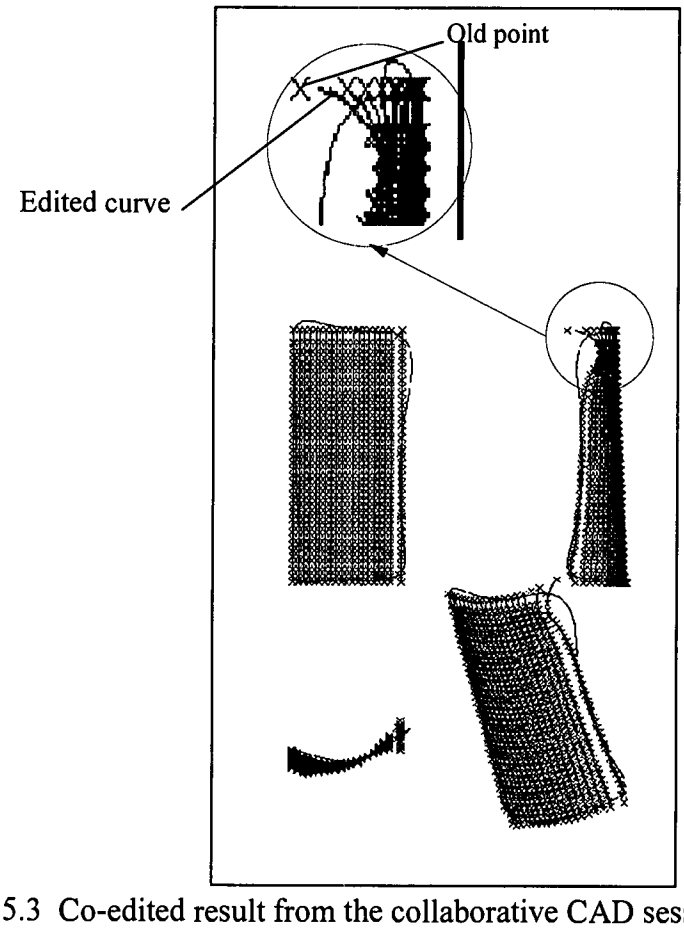


Figure 5.3 Co-edited result from the collaborative CAD session on a turbine blade

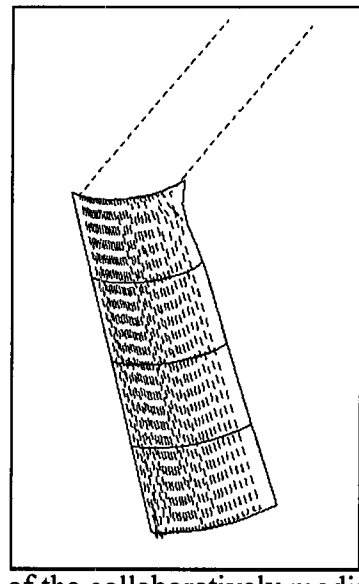


Figure 5.4 The toolpath of the collaboratively modified turbine blade surface

5.2.3 Reverse Engineering a Horn Pad through the Internet and ISDN

This section demonstrates the application of *COCAD* in reverse engineering a horn pad, a component of a steering wheel. A clay model of the horn pad, as shown in Figure 5.5, was measured by a CMM. The surface data points from the measurement were used to construct the horn pad geometry. The process to build the geometry was determined to be collaboratively operated by a measurement engineer and a manufacturing engineer in order to discard redundant measurement data and to perform necessary modification. This collaboration was determined to be assisted by *COCAD* to illustrate a reverse engineering task of a design team in fulfilling a necessary CAD-geometry modification. The data transmission through ISDN link and the Internet were explored.

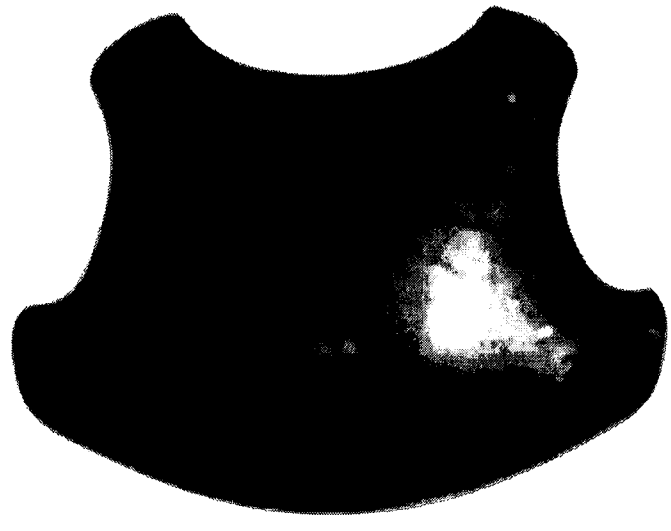


Figure 5.5 The image of a horn pad

The system layout adopted in this example is shown in Figure 5.6. This layout was revised from Figure 4.7 where SUN Sparc workstation 10s were used as the network nodes and linked through the campus LAN. The *server* and *'client #1'*, as shown in Figure 5.6, were linked through the Internet and a narrow-band ISDN connection. The ISDN connection was implemented to resolve the unreliable data transmission through the Internet (Trewitt 1990). The ISDN connection was a basic rate on-board interface which was supplied with point-to-point protocol (PPP) software. The bandwidth was 1B (64 Kbits per second) over the ISDN network of Telstra Australia. The usage of the ISDN network was similar to that of the Internet that was transparent to the user. Usage of the ISDN connection was similar to having an enhanced dial-up connection, through modem over a normal telephone line, from a personal computer to a world-wide web service provider.

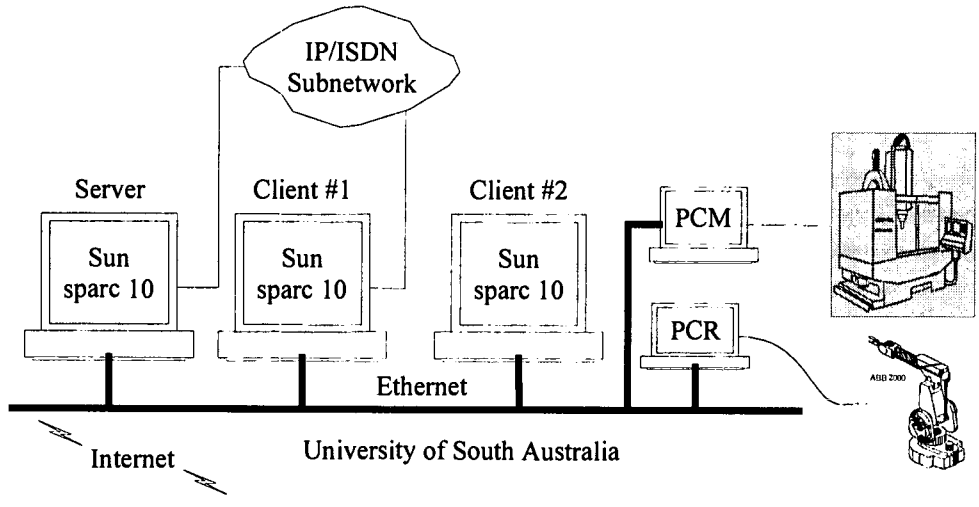


Figure 5.6 The layout of the linked system (revised from Figure 4.7)

A snapshot of the *creator's* (the measurement engineer's) screen showing both the data points and the fitted non-uniform rational B-spline (NURB) curves, are illustrated in Figure 5.7(a). The role of the *COCAD* subsystem, on the *creator's* side, was to read the data points, to reconstruct the NURB curves for the horn pad, and to transfer the NURB data to the global geometry server *ggserv* sequentially. The wireframe model of the horn pad on the *observer's* (the manufacturing engineer's) screen was composed of NURB curves, as shown in Figure 5.7(b). These curves were received by the *observer* from *ggserv*, while the *observer* was observing the operation of the *creator*. The central part of the horn pad geometry was collaboratively revised, as shown in Figure 5.7(b), through the discussion between the *creator* and the *observer*.

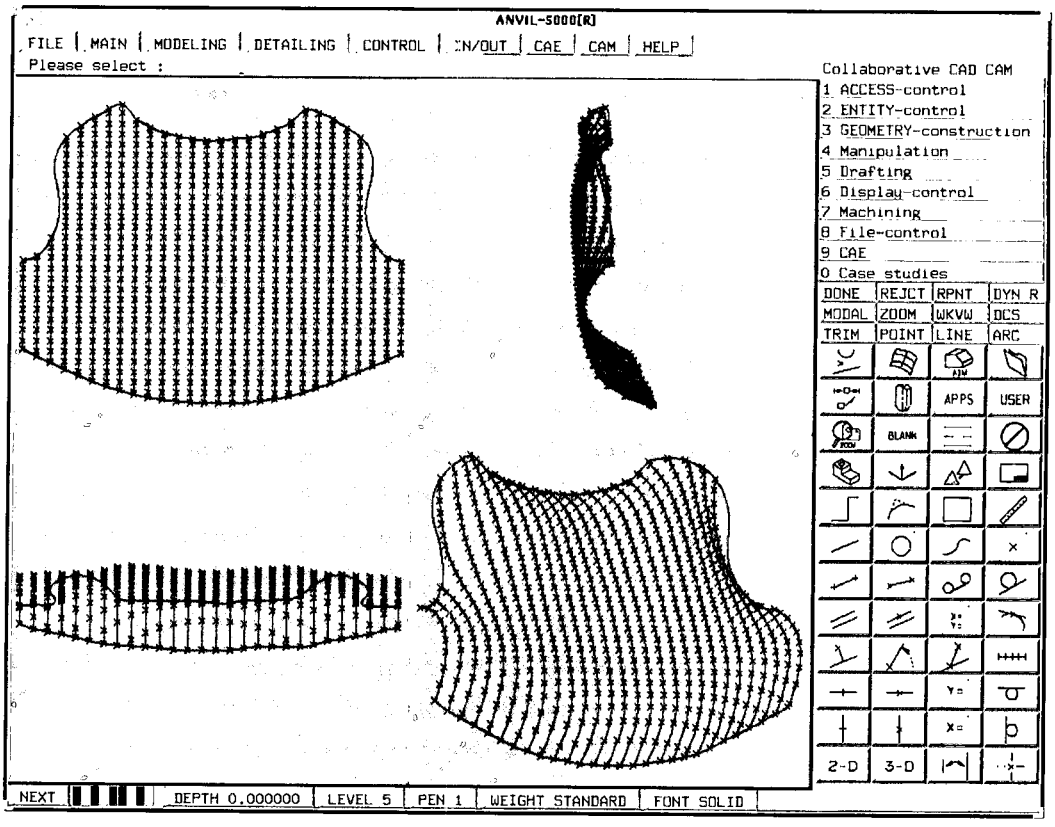


Figure 5.7(a) *Creator's* screen for horn pad coediting session

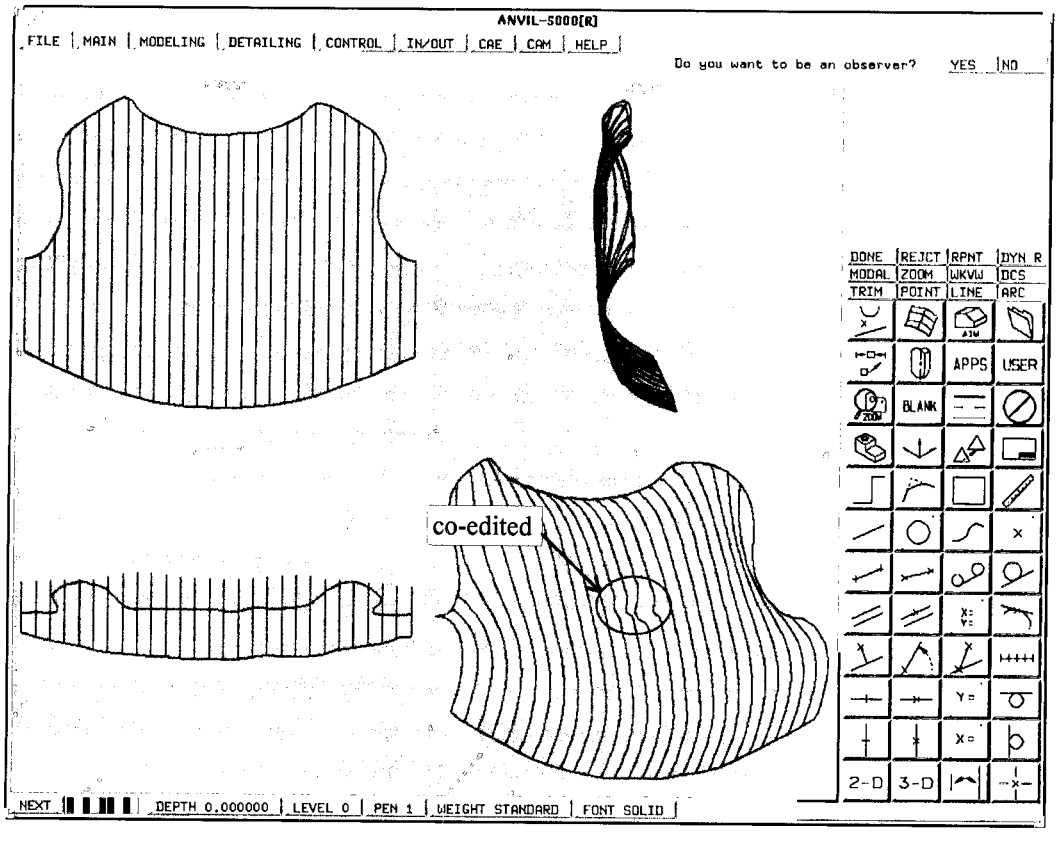


Figure 5.7(b) *Observer's* screen for horn pad coediting session (after coediting)

The human-human communication during the collaborative co-editing session was assisted by the collaborative tools such as the ShowMe and the SharedApp software (Lin et al 1995a). The collaborative tools were used to establish an information-rich communication channel. The roles of *creator* and *observer* were swapped whenever necessary between the measurement engineer and the manufacturing engineer during the collaborative session.

The data format, as defined in Table 4.1, was designed to transmit only the necessary data in minimising the network transaction time. The idea to define this data format was based on the data structure of the surface modelling, in Anvil-5000 CAD/CAM software, and the fact of unreliable Internet data transmission.

The reflection time in this example took initially more than ten seconds through the Internet rather than a few seconds as in the example of Section 5.2.2. This was not acceptable although it was expected because the Internet was publicly shared. Therefore, the data transaction algorithm adopted in the Section 5.2.2 was investigated in order to shorten the reflection time. The problem was found out that the data transaction algorithm in Section 5.2.2 transferred a data record at a time, which worked fine in local area network (LAN) but was not efficient in the horn pad example. This was resulted from the existence of twenty intermittent computer gateways, as shown in Table 2.2, in this example. The example in Section 5.2.2 was in LAN while the example in this section was through the Internet. For example, if there is a curve-meshed surface that is composed of two U curves and three V curves, the data that need to be transmitted could be '130.220.19.76 INSERT SURCM 201 2 6 4 3 200 2 20'. There are about 47 bytes in this data without counting the overhead for network transmission. Therefore, eight data transactions (as are underlined) were needed when the algorithm in Section 5.2.2 was used.

A new algorithm was determined to be developed to reduce the data transaction frequency and to shorten the network response time due to the fact that the Internet was shared by millions of users. Therefore, the data transaction algorithm was revised, in this section, to send out more bytes (data) at a time to enhance the networking efficiency. The data '130.220.19.76 INSERT SURCM 201 2 6 4 3 200 2 20' was then sent out in one transaction. Furthermore, data compression and encryption algorithm could be adopted to reduce the network data transaction but was out of the scope of this research. The transmission data size for a NURB curve that consisted of thirty points was about one

kilobytes in our test example. The network transmission time, with the revised networking algorithm, was about one to three seconds through local area network. It took about five to fifteen seconds on average through the Internet while the *ggserv* was located at the National Cheng Kung University, Tainan, Taiwan and two *COCADCAM* users were dispersed in the University of South Australia, Australia.

Furthermore, a dedicated ISDN connection was also linked to experiment the effect of a proprietary link. Although, the experimented ISDN link was only a narrow-band with the transmission rate of 64 Kbits per second, the experimental result was very promising. The response time was very stable as expected which took about one or two seconds for every transaction. The transmission time through international ISDN was expected to be very similar to the local ISDN link although the international ISDN link from Australia to Taiwan was not implemented due to the incompatible ISDN protocols commercialised by Taiwan Telecom.

The data transmission time showed that the data format, as shown in Table 4.1, and the network transaction algorithm that was revised in this example could reduce the network response time. The user's waiting time was shortened by the revised networking algorithm in this section and was very reliable through the ISDN network, as compared with the result from Kao and Lin (1996d).

In conclusion, the user's waiting time was about the network data transmission time plus one to fifteen seconds for every data transaction in the example. The network response time for data transmission is not equal to the user's waiting time since it takes computation time for a CAD/CAM package to display a CAD geometry. The computation time depends on the structure of the CAD/CAM package, the CAD file size, the complexity of the CAD geometry to be manipulated, the computer platform, and the operating system, etc. The response time, from the user's point of view, also depends on the CPU sharing (scheduling) algorithm among different processes running at the same time in a multi-tasking computer platform (Khanna 1995). These issues are beyond the scope of this research.

5.3 Application in Metal Extrusion Die Surface CAD/CAE/CAM

5.3.1 Overview

This section focuses on achieving a computerised design, analysis and manufacturing environment for calculating an optimised metal extrusion die surface, based on the *COCADCAM* system in forming a distributed and collaborative application.

Conventionally, the extrusion process was divided into three types (Altan 1983), as shown in Figure 5.8:

(i) forward extrusion where the direction of the punch movement is the same as the extruded billet;
(ii)backward extrusion where the punch movement is opposite to the extruded billet; and
(iii) combined forward-backward extrusion.

Metal extrusion was a near-net-shape manufacturing process used to produce long straight product such as gear spline, bar, wire, and seamless pipe, etc., which could reduce machining operations and increase the strength of a product.

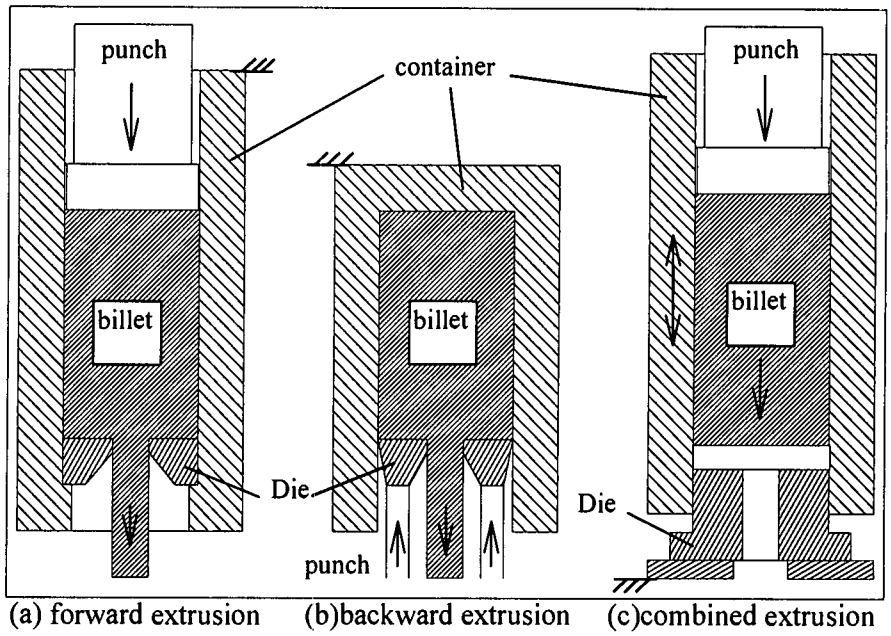


Figure 5.8 Extrusion processes

The most common shape of extrusion die was a flat-faced die profile which was easy to make but encountered a drastic transition of the metal flow from the die inlet to the outlet profiles. This drastic transition resulted in a dead metal zone which often caused severe shear friction in cold extrusion, although the boundary geometry of the dead metal zone, in hot extrusion, had the advantage of forming a natural streamlined boundary to reduce the extrusion force. The unnecessary shear friction would shorten the life of an extrusion die and increase manufacturing cost. The high shear friction was also one of the major causes resulted in asymmetrical extrusion product, therefore, post-strengthening and detwisting operations after the extrusion process were unavoidable. Thus, a better die profile, as shown in Figure 5.9, was hoped to overcome this problem. The example in this section will show an extended application of a conventional integrated procedure in metal extrusion die surface design and manufacture to a collaborative working style through the environment of the *COCADCAM* system.

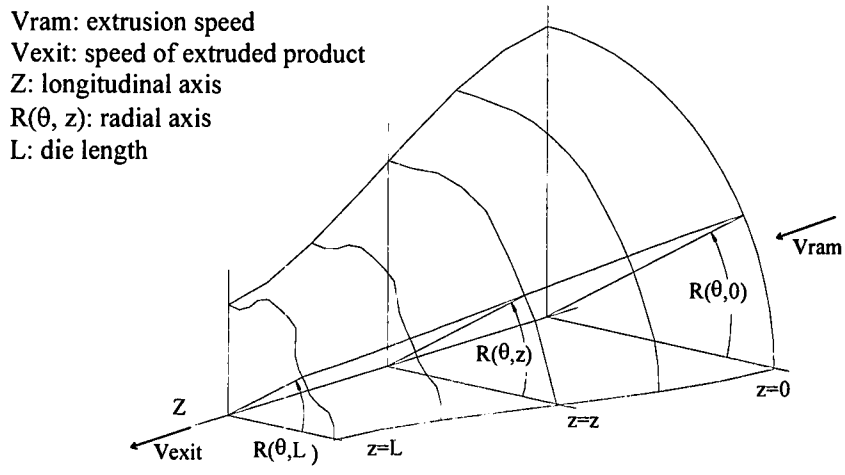


Figure 5.9 A sample streamlined extrusion die profile (from Kao and Lin 1995a)

5.3.2 Traditional Integrated Procedures

The conventional process of designing and manufacturing an extrusion die surface consisted of three sequential modules, as shown in Figure 5.10. These modules are:

(i) the selection of inlet (billet) and outlet (product) profiles;
(ii)the creation and optimisation of the die surface; and
(iii) the generation of toolpath and NC program.

The first module provides the second module, based on the user's selection, with the relevant geometry and other parameters of the product and the process, for die surface creation and optimisation. The mathematical model of the upper-bound theorem was integrated with a numerical optimisation algorithm (Kao 1988, Kao and Lin 1995a) by the second module to calculate an optimised die surface geometry. In this example, the die inlet and outlet profiles were approximated by cubic splines, and the transition curves from the die inlet to the die outlet were constructed using Bezier splines, and the die surface in the numerical optimisation was a Ferguson surface (Faux 1979). The third module controlled generation of the toolpath of the die electrode surface for the milling operation once the optimised surface was created. The NC program for the milling operation could be post-processed at the user's request.

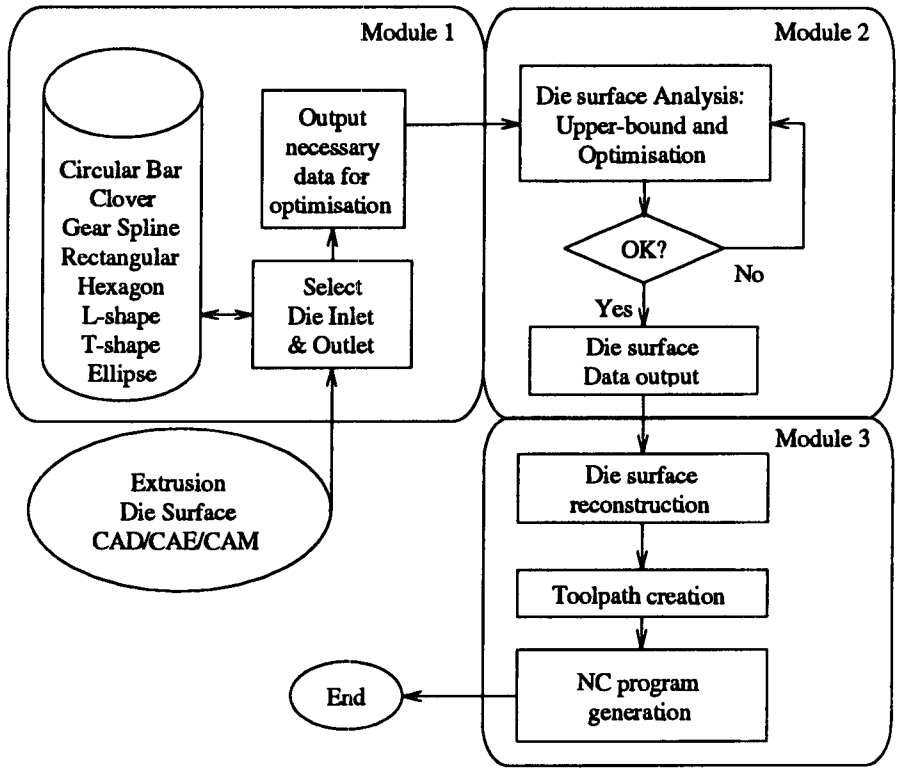


Figure 5.10 Traditional modules

5.3.3 Distributed Environment

The traditional procedures, as described in Section 5.3.2, could be tasks of a sales department, engineering design department, and manufacturing department respectively, as shown in Figure 5.11. These tasks, if properly organised, can be distributed to different geographical locations in a multi-location enterprise (Goldman 1995). For example, the sales department could be located at Australia, the engineering design department at the USA headquarters, while the manufacturing department would be better located at or near the marketplace, not necessarily in USA. Information flow among these departments can be exchanged from the sales and the engineering design department in the headquarters to the manufacturing department, through the Internet or an equivalent network. The engineering department can prepare the necessary design parameters for the die surface optimisation. The optimised surface data can be transferred to the manufacturing department for evaluating the manufacturability and the cost of making the die surface. Engineers from different departments in geographically dispersed places can *virtually* sit 'side-by-side' under a collaborative computing environment. Application of the above modules is described in the following Sections.

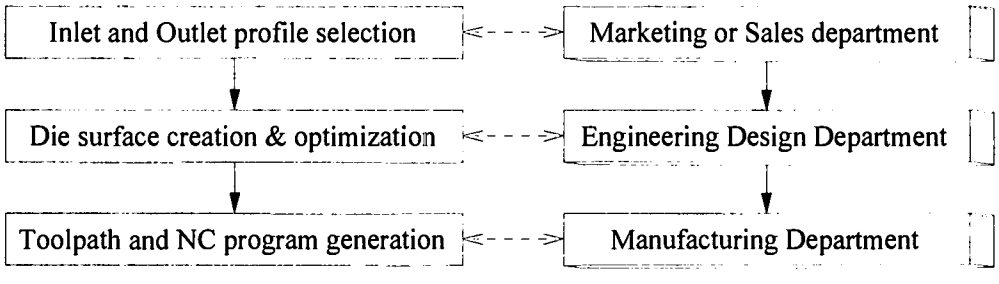


Figure 5.11 Procedures for integrated extrusion die surface CAD/CAM

5.3.4 Collaborative Design of Forward Extrusion Die Surface

5.3.4.1 Overview

This section presents the three modules, as mentioned in Section 5.3.2, in the application of a collaborative environment based on the *COCADCAM* system. The first module and the second module were assumed to be conducted by Designer A, while the third module was assumed to be operated by Designer B, for the demonstration purpose.

The graphical environment of *COCADCAM*, in Anvil-5000 CAD/CAM system, for the metal extrusion die surface design, analysis and optimisation, and toopath generation is shown in Figures 5.12(a) ~ (d). These are explained in Sections 5.3.4.2 ~ 5.3.4.4. The manufacture of the extrusion die electrode at a remote machining cell is also explained in Section 5.3.4.5.

5.3.4.2 Inlet and Outlet Profile Selection

The interactive menu, as shown in the upper-right corner of Figure 5.12(a), created in the *COCADCAM* system, was used in the selection of die outlet profiles for the 3D extrusion-die surface. An elliptic die outlet profile was selected, as depicted in Figure 5.12(a), where the billet radius was fifteen millimeters and the die length was forty millimeters. The half length of the major axis of the elliptic outlet was twelve millimeters, and the length of the half minor axis was eight millimeters. The die inlet profile was in circular shape where the diameter was assigned by the user interactively.

A data file containing the initial optimisation parameters was output once the die inlet and outlet profiles were determined. This data file was to be used by the second module, in the numerical optimisation, to calculate an optimised extrusion die surface. The first module then passed the task to the second module, as will be explained in Section 5.3.4.3, for the die surface optimisation.

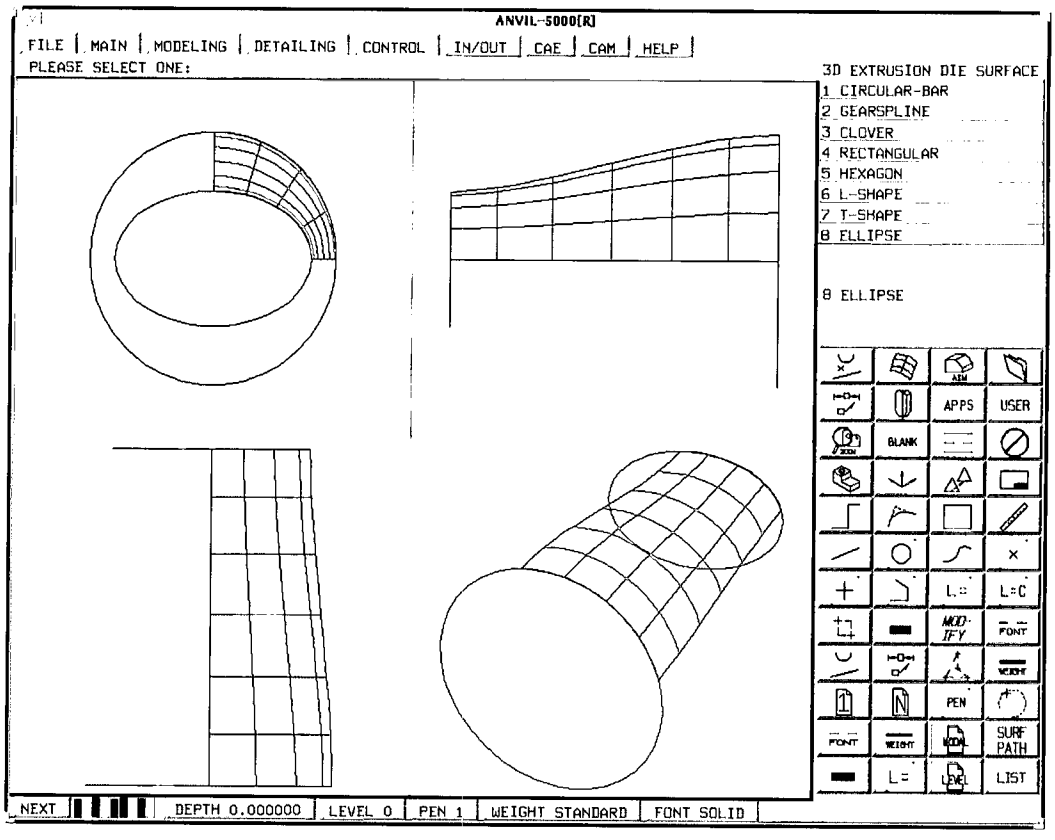


Figure 5.12(a) Profiles selection and die surface optimisation (on screen of Designer A)

5.3.4.3 Die Surface Optimisation

The optimisation of the die surface was based on the upper-bound algorithm and a numerical optimisation technique (Kao and Lin 1995a). This optimisation program (Kao 1988) was actuated by the first module to calculate an optimised die surface according to the data file output from the first module, as described in Section 5.3.4.2. It was assumed that the billet material was aluminium 1100F with the plastic flow stress $\sigma = 192.5\varepsilon^{0.208}$ and a constant shear friction factor 0.3 in the extrusion forming process.

The data points of the cross section curves in the optimised die surface were output to build 3-D curves in the *COCADCAM* system once the numerical optimisation was accomplished. These 3-D curves were displayed on the screen of Designer A, as shown in Figure 5.12(a), and transferred to Designer B, one after another, through *COCADCAM.* Therefore, Designer B can view the progress of this task, as shown in Figure 5.12(b). Once these 3-D curves were generated, a curve-meshed surface was created based on these 3-D curves to form the die surface profile, as shown in Figure 5.12(a). This die profile was again transmitted to Designer B, ready to be used by Designer B to generate the toolpath.

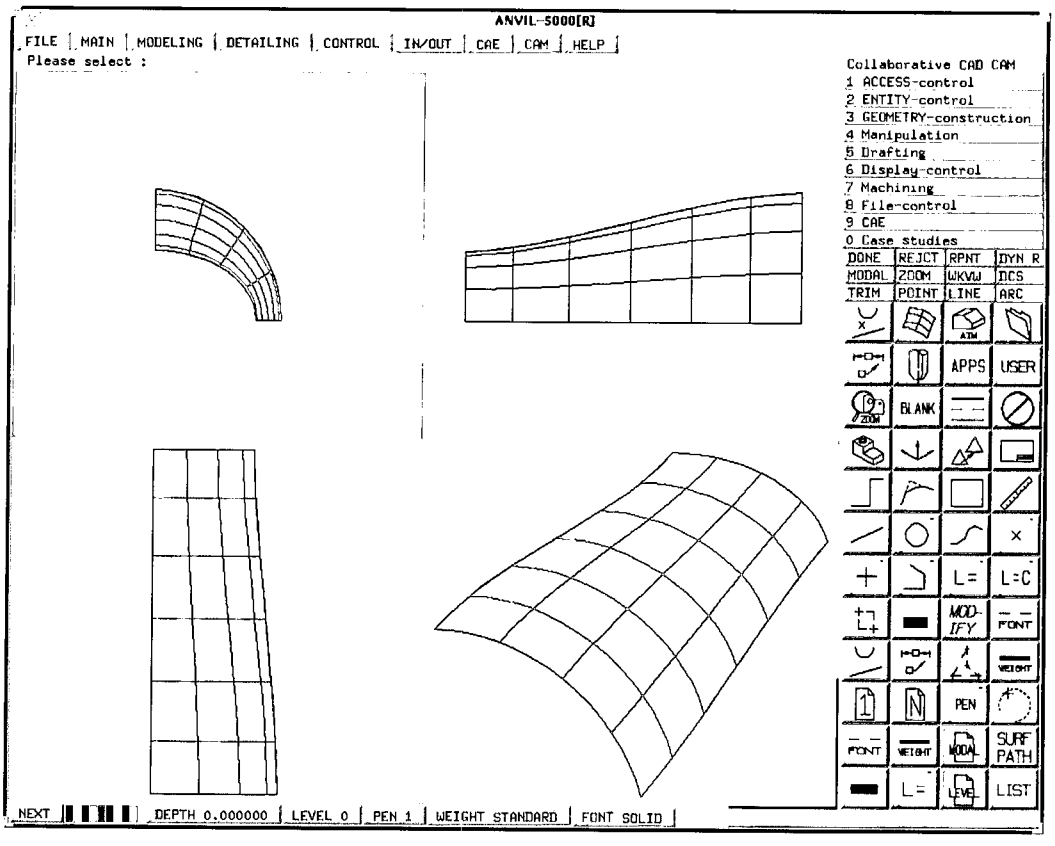


Figure 5.12(b) Optimised Die surface received from Designer A (on screen of Designer B)

Designer A was the *creator*, up to this stage, in charge of the collaborative design and optimisation of the forward extrusion die surface, while Designer B was the *observer* watching the progress of Designer A. Designer A then passed the control of the task to Designer B, since it was assumed that the third module was operated by Designer B. Therefore, Designer A became the new *observer* and Designer B became the new *creator*. Designer B then proceeded to generate the toolpath and to output the NC program for the machining task.

5.3.4.4 Toolpath and NC Program Generation

The toolpath for three-axis machining of the die electrode was created, as shown in the broken lines of Figure 5.12(c), by Designer B. The function of generating toolpath was in the *COMAN* subsystem of *COCADCAM*. The toolpath generation information was then transmitted to Designer A, as shown in Figure 5.12(d), through *COCADCAM*. Machining parameters were assigned to generate a NC program. These parameters included specifications of milling tools, machining centre, and work origin coordinates, etc. A tool

library was established according to the tools available in the tool magazine of the machining centre, which included tool material, tool dimension, and tool type, etc. Specifications of the machining centre included the spindle speed, feed rates, and working range of the machine table, etc. The work origin was pre-determined and a NC program for successive milling operations of the elliptic die surface was postprocessed according to a template file based on the above information. A machining command, as presented in Section 4.4, was sent to the server of the remote machining cell for the milling operation at a distance. The remote machining process is described in the next Section.

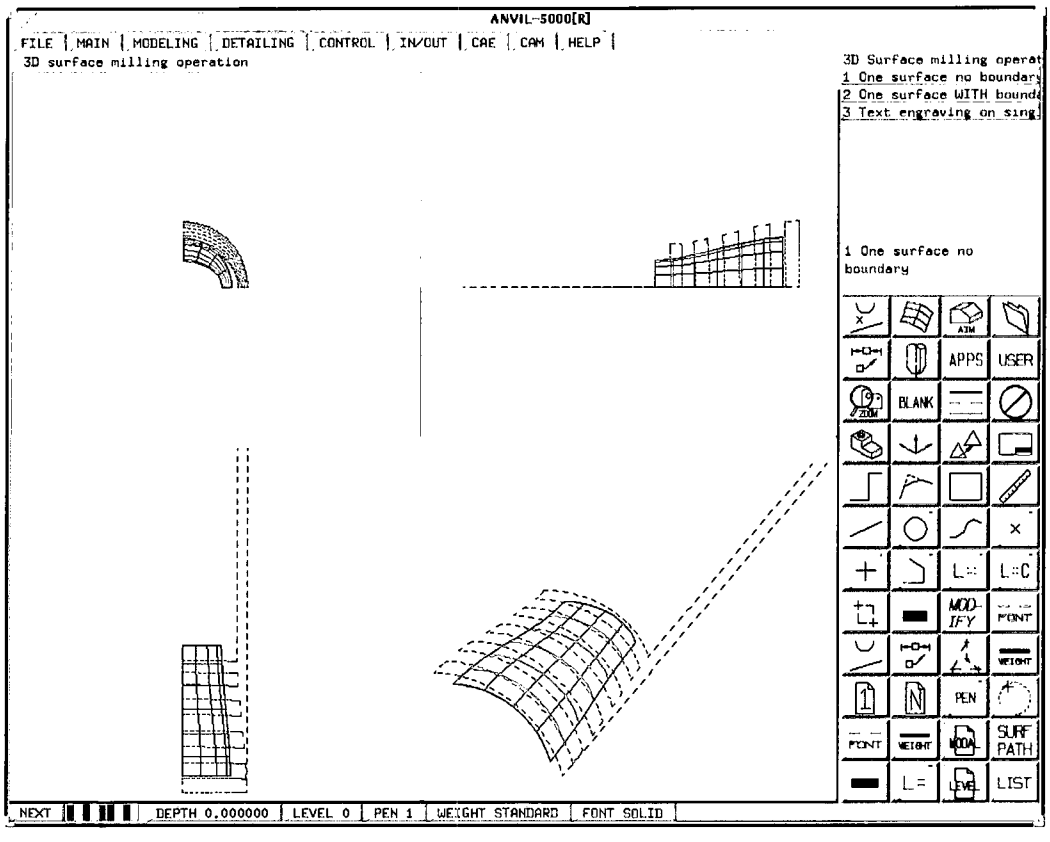


Figure 5.12(c) Toolpath generation of the die surface (on screen of Designer B)

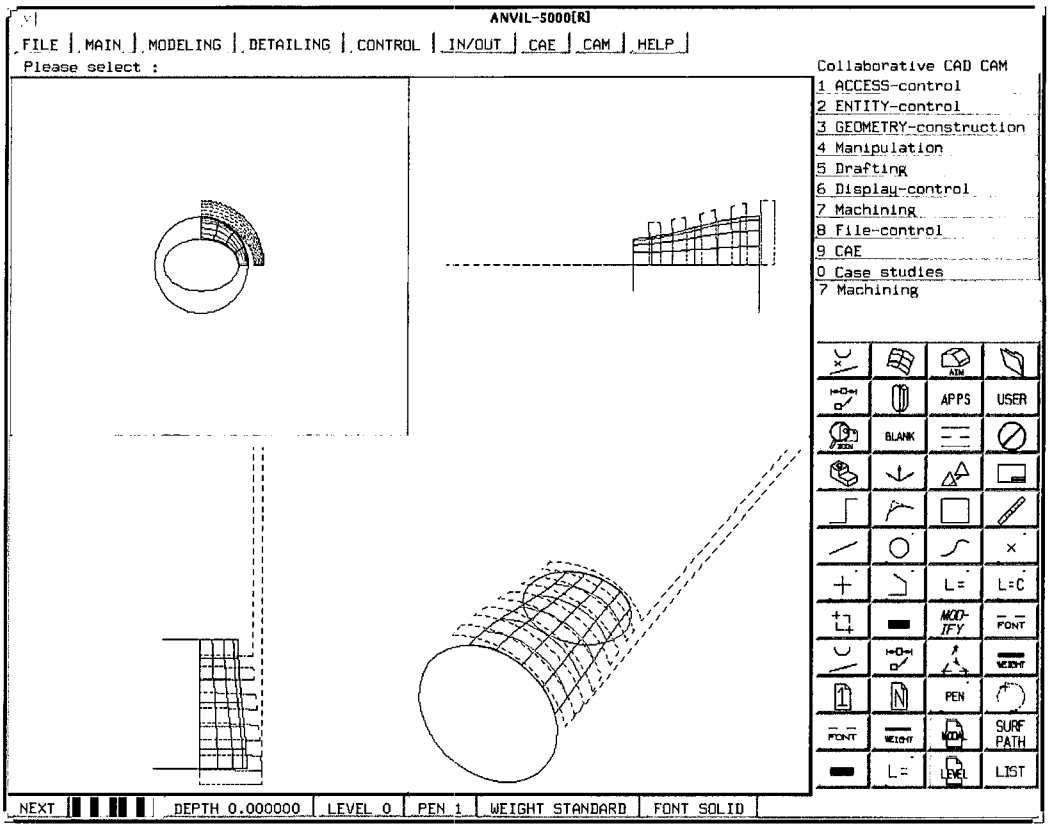


Figure 5.12(d) Toolpath received from Designer B (on screen of Designer A)

Figure 5.12(a)~(d) Graphical environment for the metal extrusion die surface CAD/CAM

5.3.4.5 Remote Machining

A remote machining cell, as shown in Figure 3.2, was used to fulfill the remote milling operation, as described in Section 3.3.4, based on the NC program output from the third module in Section 5.3.4.4.

The messages that were sent out for remote machining in this example, are abstracted and shown in Table 5.1. A message file that requested the machining centre MCV-1000CR for three-axis machining is shown in Table 5.1. The NC file name was *'O1000'*; the billet *'ALCYL1'* was a preset name of an aluminium block; and the fixture was an air-operated vice attached to the table of the machining centre. The control and monitoring program of the machining centre downloaded the NC program, whenever it was appropriate, to the CNC controller of the machining centre and *'CYCLE START'* the machining task. The control program then monitored the machining processes and would not accept any other machining command as long as it was in the operation of machining. An anomaly status would be reported from the control program of the machining centre to the server program whenever alarm occurred in the machining centre. The machined electrode with elliptic die

outlet is shown in Figure 5.13, where the surface finish of the electrode could be improved by refining the toolpath, but was not the focus of this example.

Table 5.1 Abstracted messages for remote machining

Contents:	Elliptic die
Machine	MCV-1000CR
Operation	CUT
Axis	3
NC_Filename	O1000
Billet	ALCYL1
Fixture	Vice
Date	2 Jan 1996



Figure 5.13 The machined electrode

In conclusion, this example has demonstrated that the *COCADCAM* system can be extended to the application of distributed CAD/CAE/CAM, such as the design, analysis and optimisation, and manufacture of a forward extrusion die geometry. This means the *COCADCAM* system can be customised as an environment to handle distributed collaborative task.

5.4 Application on the Generation of Preliminary Assembly Sequence

5.4.1 Overview

The significance of this example is to extend the application domain of *COCADCAM* to collaborative planning where the generation of the preliminary assembly sequence is used for illustration. The networking data was designed, in this example, to contain only necessary assembly information regarding assembly sequences rather than the complete CAD data; data transaction through network was expected to be minimised. An assembly sequence provided information about the assembly order of parts included in a product. However, the assembly sequence of a product could be affected by two major factors: 1) assembly direction and 2) the choice of a base part. This meant that the preliminary assembly sequence depended upon the choice of a base part when the assembly direction was determined. This example implemented the algorithm developed by Lin and Hsu (1996), and extended the application of *COCADCAM* to a distributed working environment for collaborative design. Geographically dispersed assembly engineers could change the base part and discuss the result of the associated assembly sequence remotely through *COCADCAM.*

5.4.2 Example of the Preliminary Assembly Sequence

This section uses an example, as shown in Figure 5.14, to explain the preliminary assembly sequence. The assembly precedence is represented in matrix form, as shown in Table 5.2, by applying *the precedence detection algorithm* and the *precedence manipulation algorithm* proposed by Lin and Hsu (1996). There are six parts in this example. For the generation of the preliminary assembly sequence, the main block (part number 6) was initially assumed to be the base part and the other parts were assembled from the upper direction relative to the base part. Therefore, the preliminary assembly sequence of the pneumatic piston assembly, as shown in Figure 5.14, was generated and is depicted in Figure 5.15. The arrowhead, as shown in Figure 5.15, means 'assembled before', for example, part 5 must be assembled before both parts 3 and 4.

A collaborative session was determined, in this research, to be needed for necessary discussion to choose a better base part and a better preliminary assembly sequence. The

reason was because a preliminary assembly sequence might not be a feasible assembly sequence. For example, the graph showing the preliminary assembly sequence, as shown in Figure 5.16, where the base part is of part 4, was not a feasible assembly sequence. This was resulted from the fact that part 3 had the same degree (same assembly priority) as the base part 4, and thus conflicted with the fundamental assumption. The fundamental assumption was that there was only one base part with the highest degree in an assembly (Lin and Hsu 1996). This demonstrated that the choice of a base part could affect the assembly sequence, therefore, the suggestion from an expert should be expected.

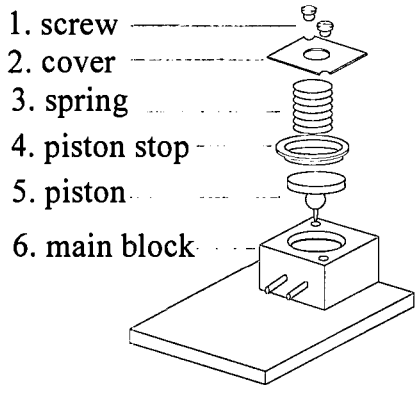


Figure 5.14 Pneumatic piston assembly (from Lin 1996)

Table 5.2 The assembly precedence matrix, base part is 6 (from Lin and Hsu 1996)

	1	2	3	4	5	6
1	0	0	0	0	0	0
2	1	0	0	0	0	0
3	0	1	0	0	0	0
4	0	1	0	0	0	0
5	0	0	1	1	0	0
6	1	1	0	1	1	0

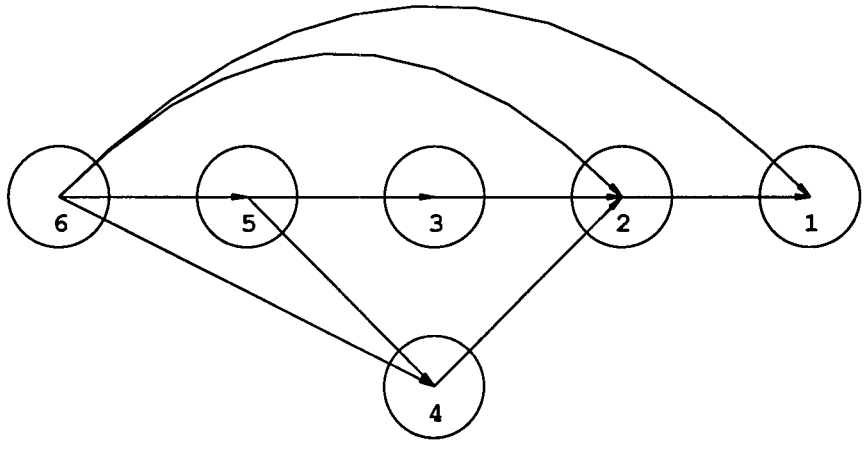


Figure 5.15 A sample graph of preliminary assembly sequence (base part 6)

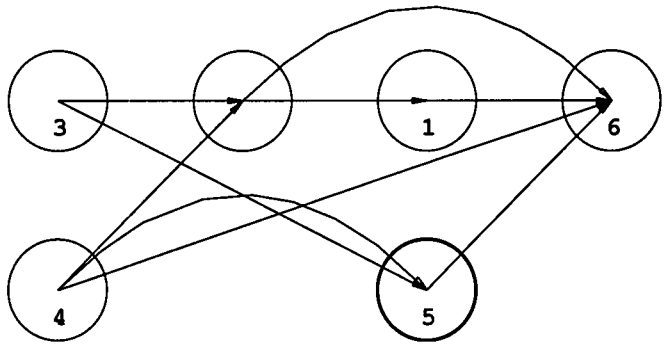


Figure 5.16 A graph of an unfeasible preliminary assembly sequence (base part 4)

5.4.3 Collaborative Preliminary Assembly Sequence Design

This section explains the application of a collaborative design on preliminary assembly sequence by using the *COCADCAM* system. The algorithm in this implementation assumed that the program evaluating the preliminary assembly sequence (Lin and Hsu 1996) had been installed on both sides, the *creator* and the *observer*. The data transaction was then customised to transmit only the fundamental assembly matrix and the necessary data so as to reduce the amount of data that needed transmission through network. There were two more customised data formats, as shown in Table 5.3, added to the *COCADCAM* system, to enable the collaborative design on preliminary assembly sequence. The first format, as shown in example 1 of Table 5.3, was used to create a new assembly sequence design. The first record, in Example 1, is the *IP address* of the sender, the second record is the *Command Name* 'INSERT', the third record is *Geometry Type* 'ASSXX' and the fourth record is the *Geometry Name* '50'. The first number in the fifth record is the total number of parts. The first number determines the total number of the other data records representing the assembly matrix. The second number of the fifth record is the number of columns of the assembly matrix. The assembly matrix was generated by the user according to the base part that was chosen, as in the third number of the fifth record. The fourth number in the fifth record is the assembly direction ('-' for negative and '+' for positive). The last number in the fifth record is the base part number which was used to display the preliminary assembly sequence graph on screen. The second format, as shown in the Example 2 of Table 5.3, shows the data format to revise the preliminary assembly sequence by changing the base part. The second record in this example is 'EDIT' and the fifth record is the base part to be changed. For example, '130.220.19.76 EDIT ASSXX 50 6' means a request from the host computer (IP address 130.220.19.76) to revise the

assembly sequence graph, that is named 50, by using the part number 6 as the new base part.

Table 5.3 Data example for collaborative preliminary assembly sequence design

	Example 1	Example 2
Owner ID (IP address)	130.220.19.76	130.220.19.76
Command Name	INSERT	EDIT
Geometry Type	ASSXX	ASSXX
Geometry Name	50	50
Data records start	10 10 9 - 8	6
...	0000000000	NULL
...	1000000000	NULL
...	...	NULL
Data records end	0000000000	NULL

For the purpose of illustration, an alternator assembly, as shown in Figure 5.17, will be used to demonstrate a collaborative design session on the generation of preliminary assembly sequence. The collaborative session starts from a precedence matrix which is assumed to have been determined by the designer according to the algorithm proposed by Lin and Hsu (1996). There are fourteen parts in the alternator assembly, as shown in Figure 5.17, and the initial precedence matrix is shown in Figure 5.18. A simplified matrix is shown in Figure 5.19(a) where group 6 (part 6, 7, 8 and 9) and group 13 (part 12 and 13) are determined to be subgroups. The algorithm in determining these two subgroups can be referred to Lin and Hsu (1996). To simplify the graph display of the preliminary assembly sequence, the matrix in Figure 5.19(a) was reorganised as in Figure 5.19(b). The data that are transmitted through network, in this example, for the collaborative design are shown in Table 5.4.

A snapshot of the preliminary assembly sequence graph, according to Table 5.4, is shown in Figures 5.20(a) and (b). The *creator's* screen layout is shown in Figure 5.20(a) where the base part ⑨ was selected by the *creator*. The *observer's* screen, as shown in Figure 5.20(b), then showed the same result as the *creator's*. However, part ⑨ was a screw and was apparently not appropriate to be used as the base part. Therefore, the *observer* used text 'talk' program, as illustrated in the middle-left window of Figure 5.2, to discuss synchronously with the *creator*, and suggested the base part to be changed from ⑨ to ⑥ (front housing). The *observer's* screen was then illustrated in Figure 5.20(b) after the *creator* finished the change by selecting the sub-function 'Assembly precedence' from the

fourth function 'Manipulation', as shown in Figure 4.8. The data transaction that made this change is shown by the example 2 in Table 5.3. To change the base part, the *observer* could alternatively ask the *creator* to drop the control token, and then took the control and did the change.

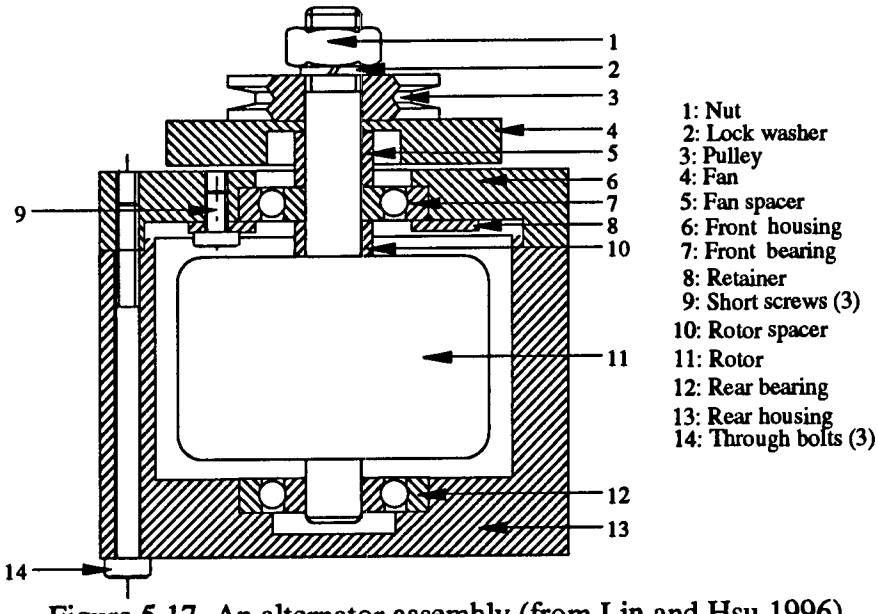


Figure 5.17 An alternator assembly (from Lin and Hsu 1996)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1														
2	1													
3		1												
4			1											
5				1										
6							1	1	1					1
7					1			1						
8									1					
9														
10							1							
11	1	1	1	1	1		1			1				
12											1			
13						1						1		1
14														

Figure 5.18 The initial precedence matrix (from Lin and Hsu 1996)

	1	2	3	4	5	6	10	11	13	14
1										
2	1									
3		1								
4			1							
5				1						
6					1					1
10						1				
11	1	1	1	1	1	1	1			
13						1		1		1
14										

(a) The simplified precedence matrix

	1	2	3	4	5	6	7	8	⑨	10
1										
2	1									
3		1								
4			1							
5				1						
6					1					1
7						1				
8	1	1	1	1	1	1	1			
⑨						1		1		1
10										

(b) The resorted precedence matrix for display

Figure 5.19 The simplified precedence matrix (base part 13 or ⑨)

Table 5.4 Data transaction example for Figure 5.19(b)

Owner ID (IP address)	130.220.19.76
Command Name	INSERT
Geometry Type	ASSXX
Geometry Name	50
Data records start	10 10 9 - 9
...	0000000000
...	1000000000
...	0100000000
...	0010000000
...	0001000000
...	0000100001
...	0000010000
...	1111111000
...	0000010101
Data records end	0000000000

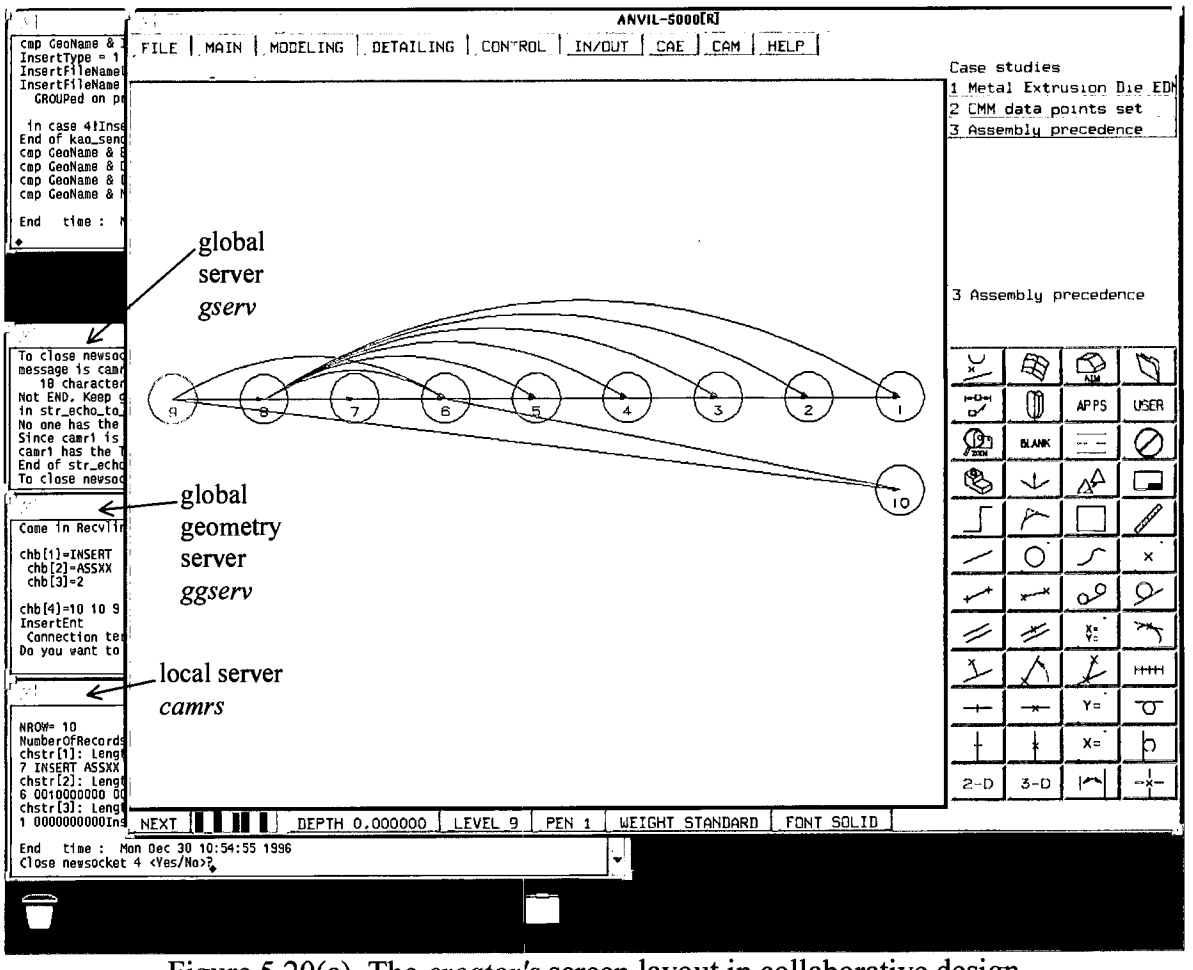


Figure 5.20(a) The *creator's* screen layout in collaborative design

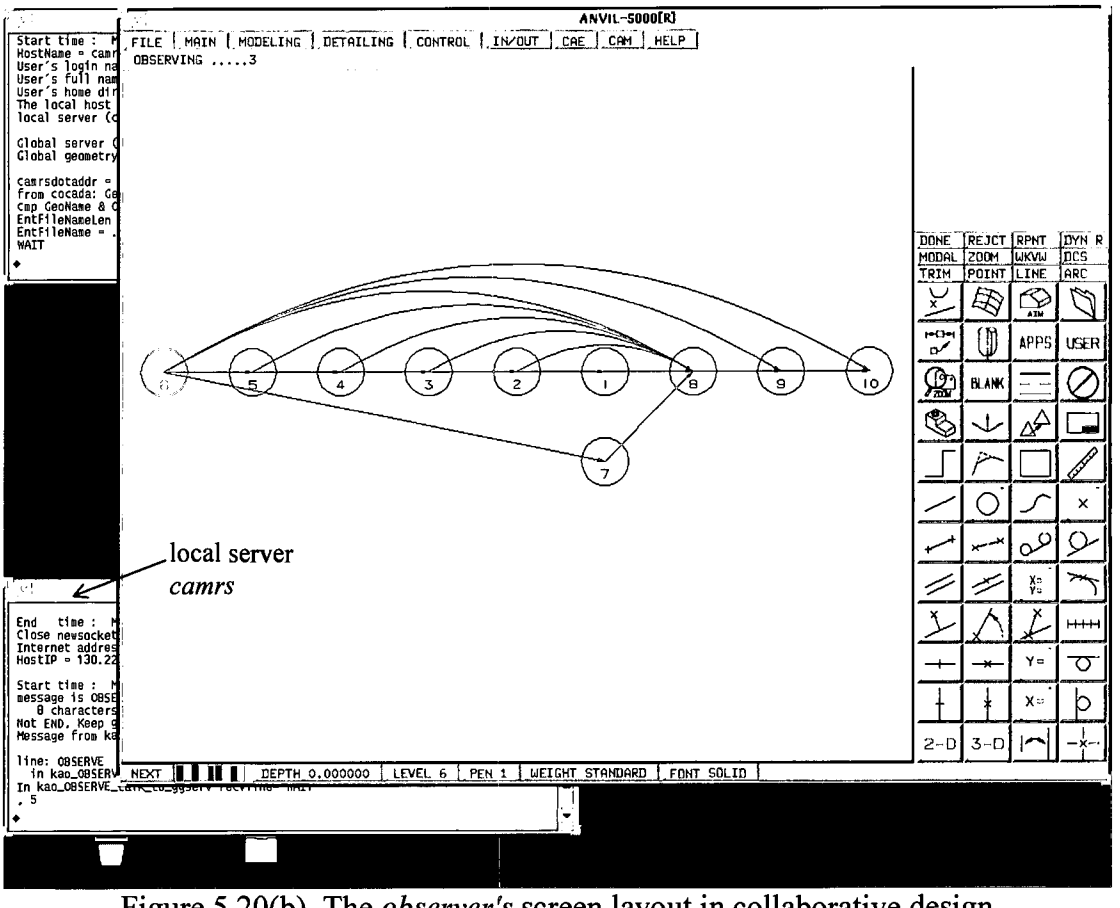


Figure 5.20(b) The *observer's* screen layout in collaborative design

The data format that was defined and explained in Section 5.4.3 for network transaction could reduce the data in network transaction for the purpose of displaying the preliminary assembly sequence graph. For example, there were 54 entities (including 10 circles, 10 texts, 17 curves and 17 arrow heads) as shown in Figure 5.21 and Figure 5.22. These 54 entities were grouped and represented by one geometry name for identification. This data format made the network data transaction algorithm adopted in this section more efficient than that in Section 5.2.3. The data transaction algorithm used in Section 5.2.3 transferred all the geometry data while this section transmitted only the fundamental data instead of all those 54 entities' data; therefore, the preliminary assembly graph could be updated as a whole. The average reflection time for this example was 12 ~ 30 seconds in the weekday and 12 ~ 17 seconds in the weekend. The time periods were measured while the server programs, *gserv* and *ggserv*, were both located in Taiwan on a host computer springsoft.com.tw (IP address 203.66.26.145).

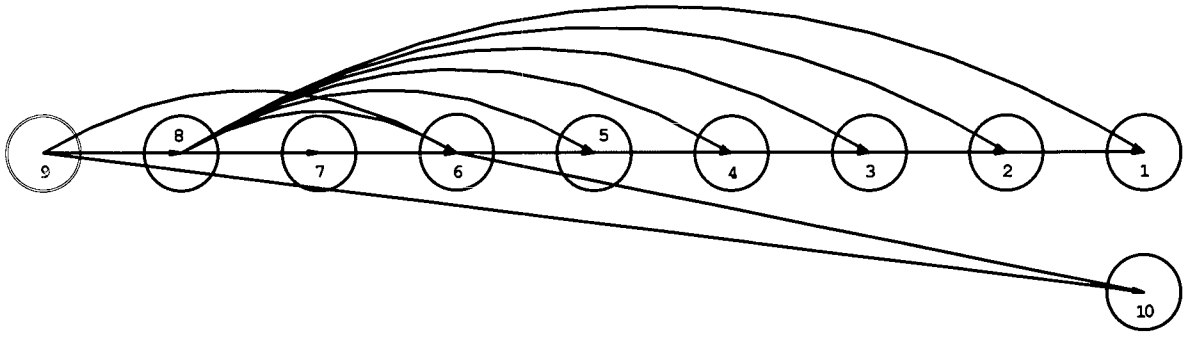


Figure 5.21 Graph of preliminary assembly sequence (abstracted from Figure 5.20(a))

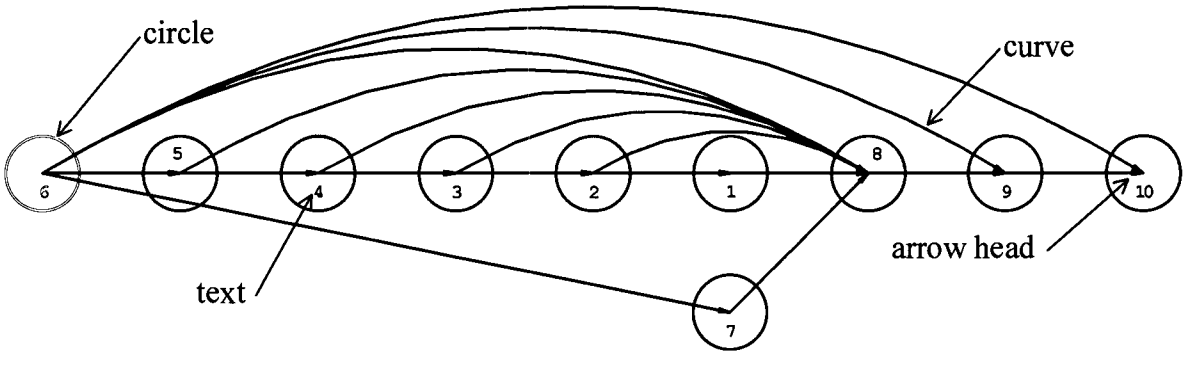


Figure 5.22 Graph of preliminary assembly sequence (abstracted from Figure 5.20(b))

5.5 *Concluding Remarks*

The *COCADCAM* system has been used as an enabling tool in several examples. The example in reverse engineering, as illustrated in Section 5.2, showed that *COCADCAM* could assist the collaborative tasks of co-editing CAD geometry. Geographically dispersed CAD/CAM users could co-edit CAD geometry as long as they were networked through either the local area network or the Internet on top of the TCP/IP protocols. The local link through ISDN was tested and the data transmission rate was very stable as expected.

In the example of metal extrusion die surface CAD/CAE/CAM, as illustrated in Section 5.3, the *COCADCAM* system was firstly used to input the die inlet and outlet profiles serving as a graphical user interface environment. A data file keeping the initial optimisation parameters for the die surface optimisation was then output before *COCADCAM* notified the optimisation program, the second module, for the optimisation. The optimisation program was a C program that was linked to the *COCAD* subsystem of *COCADCAM*, which was arranged to be actuated by the first module. The optimisation program output the optimised cross section data points of die surface, based on the input data file. *COCADCAM* then built the cross section curves and the die surface geometry, on the *creator's* (Designer A's) screen. The cross section curves were transmitted to the *observer* (Designer B) one after another through the *COCADCAM* system environment. The die surface was also transferred so that the same result was shown on the *observer's* screen for the purpose of collaborative discussion. Both *creator* and *observer* could then make necessary modifications at a distance if they wish. The toolpath and the NC program was then generated by Designer B after the *creator* (Designer A) passed the controlling token to Designer B. The *COCADCAM* system then linked and notified the server of the remote machining cell for the machining operation of the die electrode surface if Designer B wished to do so. The status of the remote machining cell could also be dynamically accessed (Kao and Lin 1997a) through the interface of a general World Wide Web browser, for example, Netscape Navigator and Internet Explorer. Thus, the collaborative CAD/CAM system *COCADCAM* became a distributed collaborative CAD/CAM with a machining cell that was operable through the Internet or equivalent network.

The last example, as explained in Section 5.4, used *COCADCAM* to assist the collaborative planning on the preliminary assembly sequence design. There were two new data formats defined, for the preliminary assembly sequence design, to reduce the network

data transmission time. The assembly sequence graph was considered as a group in these new data formats. This was based on the assumption that the evaluation program, on the preliminary assembly sequence, was linked to *COCADCAM*. Under this assumption, the data transaction for the preliminary assembly sequence, as shown in Figure 5.21 that consisted of 54 entities, only had about 137 bytes, as depicted in Table 5.4. The assembly sequence graph, as shown in Figure 5.22, was revised from Figure 5.21 by changing the base part from part ⑨ to part ⑥. The data transaction for this revision needed only to send around 28 bytes, that is '130.220.19.76 EDIT ASSXX 50 6'. With these new data formats and networking algorithm, the network response time on the collaboration of the preliminary assembly sequence design was almost the same as the co-editing of interpolated NURB splines.

In conclusion, the experimental study of the *COCADCAM* system, developed in this research, has been conducted. The idea was to use *COCADCAM* to facilitate collaborative work and the results were as expected. On the one hand, a dedicated communication medium ISDN was implemented and tested demonstrating promising consequences; on the other hand, customised data formats were designed to shorten the network transmission time, that is, to enhance the collaborative display reflection.

Chapter 6 Discussion

The development of a collaborative CAD/CAM system and its applications have been presented in this thesis. This development focused on the CAD/CAM environemnt and extended the single-location CAD/CAM technology to multi-location application for CAD/CAM collaboration.

This research started from the exploration of the existed collaborative tools linking geographically distributed locations in a realistic Internet connection between the University of South Australia (UniSA) in Australia, and National Cheng Kung University (NCKU) in Taiwan. This was an important exploration because the network communication was an inter-continental connection between Australia and Taiwan; the networking routes for the data communication went through twenty computer gateways, as shown in Table 2.2, across the United States and Japan. This exploration found that the Internet communication between Australia and Taiwan was unreliable for synchronous CAD collaboration; the response time that was required to access a remote file, through the network file system (NFS), fluctuated very much depending on the Internet traffic. The Internet was shared by millions of users all over the world and a general user did not have any control on the networking priority. Therefore, a dedicated communication link was sought, which resulted in an integrated services digital network (ISDN) to be implemented to test the reliability and effectiveness of a direct dedicated connection. As expected the ISDN implementation showed very reliable data transmission rate. However, the implemented ISDN was a narrow-band 64kbits communication medium due to the limitation of the adopted Sun SPARC 10 workstation. Faster networking media such as Asynchronous Transfer Mode (ATM) and Frame Relay (Barron 1996, Cascade 1996) are expected to provide better transmission rate.

A feasible framework was implemented using both the Internet and ISDN to enable two or more specialist CAD/CAM engineers to work together as a team on a three-dimensional computer-aided design so that each partner could provide synchronous contributions. This possibility resulted in an application environment for international CAD/CAM collaboration beyond physical barriers. A product was able to be designed and analysed on one side of the world, and manufactured on the other, immediately.

Conventional CAD/CAM collaboration was supported by using collaborative tools to establish a computer-supported cooperated working environment. Examples of these tools could be:

1) text-based messages such as text e-mail, text talk, and facsimile;

2) audio-based messages such as voice mail, telephone and Internet phone;

3) two-dimensional drawings such as blueprint drawing by facsimile and mail (includes express delivery);

4) asynchronous computer support such as multimedia email, file transfer (FTP) and world-wide web (WWW) services; and

5) synchronous computer support including shared applications, multimedia audio, whiteboard and video, Internet (video)phone, 3D geometry viewers, and desktop conferencing.

These collaborative tools were very useful in assisting the communication between geographically dispersed collaborators, which had been demonstrated in Section 2.2. However, these tools came from different vendors, therefore, one single tool that can provide the functions above is expected.

A new function *'Hot-Link'* was implemented for a traditional single-location CAD system, Advanced Solid Modeller (ASM), to gain the advantages provided by collaborative tools and network resources, which was described in Chapter 2. This function was proposed to keep up-to-date reflection of design changes, after extensive discussions and acquired the experience in conducting the international link of the CAD/CAM collaborative work.

A remote machining environment was explored, which showed the feasibility to extend CAD collaboration for collaborative CAD/CAM, by linking a remote machining facility. Thus, a remote machining cell (RMC) was implemented accordingly, as described in Chapter 3, to enable a remote machining function in the extension of the developed collaborative CAD/CAM system. This remote machining cell consisted of one four-axis

machining centre for the machining task and one robot for the workpiece loading and unloading tasks. The RMC could be actuated by the *COCADCAM* users for the machining task at a distance. The machining status could be monitored audiologically through any JAVA enabled Web browsers, such as Netscape Navigator and Internet Explorer (Kao and Lin 1997a).

The network communication structure of the remote machining cell (RMC), as described in Section 3.4, was based upon the Windows 3.11 and Sun Solaris environment. This communication was based on the network file system (NFS) protocols. With the TCP/IP network driver became standard and supported by various operating systems, such as Windows 95, the communication structure of the remote machining cell could be constructed by installing software, such as Samba (Samba 1997), on the Sun Solaris environment. Samba was a suite of programs which work together to allow clients to access to a server's file space and printers via the SMB (Server Message Block) protocol.

There were two methods that could be adopted to establish a computer-supported cooperative environment for CAD/CAM collaboration, based on existed network and computer resources. One was the assistance by collaborative tools, where new functions must be developed for CAD/CAM software to take advantage of the available computing resources. The adoption of only collaborative tools was insufficient for dynamic and interactive international CAD/CAM collaboration. Therefore, the extension of single-location CAD/CAM software to multi-location application was necessary for effective CAD/CAM collaboration. The other was the extension of CAD/CAM software with network functions for network data communication. Collaborative tools could still be used to enhance oral and text communication. These methods had been explored and resulted in the development of a collaborative CAD/CAM system (*COCADCAM*) which was described in Chapter 4.

The *COCADCAM* system could be applied for collaborative work between CAD/CAM users, as was illustrated in Chapter 5. The collaborative CAD/CAM system *COCADCAM* was implemented in this research, which extended a single-location CAD/CAM software to a multi-location application; the designers at geographically dispersed locations could co-edit CAD/CAM geometry data collaboratively at a distance. A new product that was designed on one side of the world, could be machined on the other side through the *COCADCAM* system linking to a remote machining cell. *COCADCAM*

was also extended to and applied for a collaborative planning application, on the generation of the preliminary assembly sequence, between distributed engineers working on the design for assembly. Other application examples were: revision of CAD geometry on reverse engineering a physical component (Section 5.2); and distributed CAD/CAE/CAM integration on metal extrusion die surface optimisation and machining (Section 5.3). A remote machining cell was also implemented to perform the milling machining task of the die electrode surface at a distance apart which was described in Section 5.3.4.5.

The external networking modules, as shown in Figure 4.17, were the key to the CAD network data communication in *COCADCAM*. The system configuration, as depicted in Figure 4.2, needed modification to adapt the functionality of different CAD/CAM application programs. The local client process *camrl* was not needed, if the API, for example, *GRAPL-IV* language of Anvil-5000 or GRIP language of Unigraphics II, provided networking subroutines, for the collaborative CAD system *COCAD* to link local server *camrs*. This was due to the fact that *camrl* was serving as a bridge between a CAD/CAM application and *camrs*. Moreover, if the API had subroutines to be used as a network server, the existence of *camrs* was not necessary. Both *camrl* and *camrs* were modules of the external networking modules.

The limitation of an existed single-location CAD/CAM application was resolved for the extension to a multi-location application. A traditional single-location CAD/CAM package was adopted in exploring the necessary functions to construct a multi-location CAD/CAM application. The UNIX interprocess communication, the socket-based connection-oriented server and client model, and the network file system (NFS) were implemented as the basis for the data communication in *COCADCAM*. The geographically dispersed designers and engineers could work together, through the proposed configuration of *COCADCAM*, as a team. For example, a designer on one side of the world could work with the other designer on the other side of the world simultaneously for a common design activity.

The capability of the *COCADCAM* system was limited by the traditional single-location application, although an exciting advancement had been achieved by the proposed strategy for collaborative CAD/CAM application. The CAD/CAM system that was adopted did not support multi-user interface features such as multiple mouse cursors. The

interactive operation, that was provided by the *API* functions in Anvil-5000 software, could not accept messages from external programs. The *API* program needed to communicate with the external program for the data and message exchange through the interprocess communication (IPC). This was necessary because the *observer* needed to observe the *creator's* drawing messages at all times, during the collaborative session.

The data base structure of the single-location CAD/CAM was not designed for multi-location application. The geometric entity or derived entity in Anvil-5000 was assigned a sequence number automatically once the entity was created. The sequence number was used as a pointer to keep track of the entity relationship in Anvil-5000, which was neither changeable nor reusable. If a curve was created after a curve that had the same attribute was deleted, the sequence number was different. Therefore, a special attribute, the entity name, was defined to trace the entity for geometry modification in *COCADCAM.* Although the entity name was not a standard attribute of a CAD geometry, it had been adopted and used successfully in the implementation of the *COCADCAM* system.

A new data exchange format was proposed in *COCADCAM,* as shown in Table 4.1, for effective and dynamic CAD/CAM collaboration, so as to transfer necessary data for collaborative CAD geometry manipulation and extended application. Neutral data exchange standards such as Initial Graphics Exchange Specification (IGES) and Data eXchange File (DXF) formats were not designed for co-editing CAD geometry but for the data exchange of the whole CAD file. Conversely, *COCADCAM* exchanged only the minimum data for necessary CAD data modification. For example, if there was only one point needed to be changed in a curve that was composed of 100 points, it was not necessary to exchange all the 100 points but the index of the modified point and its new coordinates. This was very helpful especially in the minor revision of a CAD file that normally had several megabytes in size. However, it was not the author's intention to create new CAD/CAM data exchange format but to disclose the deficiencies of existing neutral format in CAD/CAM collaboration and to implement the *COCADCAM* concept.

Chapter 7 Conclusion and Future Direction

This research has implemented a collaborative CAD/CAM system (*COCADCAM*) for CAD/CAM collaboration. This system was the first system in extending a single-location CAD/CAM to multiple-location application. The essential implementation strategies have been proposed and realised, and further study is inspired to enhance the *COCADCAM* system.

The *COCADCAM* system has been implemented between two geographically dispersed CAD/CAM users. Further development is envisaged to enable multiple designers for collaborative CAD/CAM operation, such as database management, concurrency control, collision control and resolution, and access control. The recovery of the system after an unexpected crash during collaboration needs to be explored. Concurrency control must also be enhanced to eliminate waiting time between processes during the collaboration.

World-wide trends of the long-existed global manufacturing philosophy are forcing us to utilise fully the best available equipment, technology and human resources to improve competitiveness. This research has presented a successful development and implementation of a multiple-location CAD/CAM system for CAD/CAM collaboration that is extendible for remote machining under the Internet and the ISDN or other equivalent networks. This system can be adopted to reduce the lead time in global manufacturing.

COCADCAM could be used as a tool to facilitate geographically separated designers forming a design task force for multiple-location collaborative product development. The geographically dispersed designers could co-edit the three-dimensional CAD geometry such as NURB splines and surfaces. A tool path could be generated once the collaboratively discussed surface was finalised. A numerical control (NC) program for the free form surface could be post-processed if requested after the generation of a tool path, which could then be used for remote machining through *COCADCAM*.

Worldwide expertise, technology and resources could be linked through *COCADCAM*. This linkage could use accessible international network facilities to establish a 3D CAD/CAM design environment for geographically distributed product designers. The

limitations of existed single-location CAD/CAM technology was resolved for multiple-location application through the implementation of the networking strategy adopted in this research. The local area network, Internet and ISDN had been experimentally tested and shown the performance as expected. A CAD/CAM engineer could work on a common design and manufacturing activity with the other remote engineer collaboratively and interactively through the *COCADCAM* system implementation.

New algorithm could be explored to further shorten the network data transmission time. In this research, two methods were successfully adopted to tackle the unreliable data transmission problem of the Internet. One was the adoption of a narrow-band ISDN to form a dedicated data transmission connection. The other was to reduce the data by defining the data format for the network transaction to transfer only the necessary data instead of the whole CAD geometry. These were presented in Chapter 4 and Chapter 5. However, the proposed data format was not comprehensive but to validate the feasibility and effectiveness of a dynamic network data transmission for interactive CAD geometry co-editing.

Data transmission speed through the network could be further enhanced by revising the communication protocols that were defined in the collaborative CAD/CAM system *COCADCAM*, although useful progress had been achieved in this research. The data format could be revised according to the characteristics of CAD geometry and the purpose of the collaboration. Data compression techniques could also be adopted to further reduce the amount of the data in network transmission.

Further development is necessary for the remote machining of multiple-surfaces. The *COCADCAM* system could be used for co-editing unlimited NURB splines. The NURB surface was based on NURB splines forming a curve-meshed surface and then converted to a NURB surface for the collaboration. Only one surface could be used at a time to generate a tool path and related NC program for remote machining in the proposed *COCADCAM* system.

The mechanism of dynamic data exchange (DDE) under the Microsoft Windows operating system was very helpful in exchanging data among different applications within a PC. If this mechanism could be extended for the data exchange among heterogeneous computer platforms, collaborative work as proposed in this research can be further

enhanced. The data format could be based upon DXF, IGES, STEP or other feasible standards.

Agent Communication Language could be used to enable the remote machining cell (RMC) to accept machining tasks from remote site through network. The remote machining cell (RMC) adopted home-brew data protocols and networking programs; so, although it could accept commands through the Internet, it spoke a dialect. Agent Communication Language, such as Knowledge Query and Manipulation Language (KQML) (Finin 1994) and Knowledge Interchange Format (KIF) (Genesereth 1992) could be adopted to extend the RMC to be an RMC agent (ARMC). If KQML and KIF could be used for the communication among disparate programs, ARMC would then become an open system agent (KQML-speaking program) that could communicate with other agents, via KQML messages, and accept milling commands from remote sources. However, the application of the agent communication technology in the remote manufacturing field is still a new research area.

A direct numerical control (DNC) function based on DNC2 needed to be explored. Currently, the remote machining cell controlled and monitored a machining centre from a personal computer over the DNC2 interface. A NC program needed to be downloaded into the memory of the CNC controller before this NC program could be cycle-started. Therefore, the file size of the NC program was limited by the free memory space of the CNC controller. The development to construct the DNC function through DNC2 is therefore needed to resolve the file size limitation. The remote machining cell can then be used for complex multi-surface machining which usually has very large NC file.

A NC program template could be created on the CNC controller to act as the DNC program for machining a large NC file. This template could consist of necessary macro variables and conditional routines waiting for the three dimensional coordinates to come in from the controlling program on the PC through the DNC2 interface. Only one NC program is allowed to be executed in the DNC2 mode. The PC controlling program needs to interpret the NC program and sends the data to CNC controller's macro variables addresses. One or more flags need to be set for the timing control so that the whole NC program does not have to be downloaded to the CNC controller before the machining operation could be started.

The CAD/CAM software can have a module to become plug-ins of WWW browsers and viewers (navigators) in order for the WWW to communicate CAD/CAM data. However, WWW clients should have the capability to talk to the other WWW clients instead of talking only to the WWW servers. The CAD/CAM application can be the user interface for the concurrent kernel of co-editing CAD and related data.

The virtual reality modelling language (VRML) can be explored and extended for the co-editing function. The 3D geometry viewing (Bell 1995, Kress 1996, ISO 1996) could only assist the collaborators to conduct collaborative work but was unable to modify the CAD geometry. Another solution would be to enable the CAD/CAM software to become a CAD/CAM application that is aware of collaborative operation.

Further investigation can be the exploration of collaborative interaction on engineering design work practice, effect of time delays, time to reflect interactive design change, linkage to expert knowledge base for optimum design recommendation, and enhancement of collaborative computing tools. Also the exploration of the collaborative algorithm if both CAD/CAM users are *creators* instead of one *creator* and one *observer*.

The traditional computer integrated manufacturing (CIM) system can be extended in forming a link among geographically separated locations in forming a virtual CIM environment with the incorporation of the computing and networking technology adopted by the *COCADCAM* system. The environment of the virtual CIM can transcend geographical boundaries and embrace the global application of CIM for enterprise integration.

Potential benefits from the *COCADCAM* system application will be the reduced lead time to market, the production of the right product in the right place, elimination of product shipment time, increased competitiveness and profitability for global manufacturing. Multiple-national companies can arrange available worldwide expertise needed by a project by locating the project team members in different countries (at an international level) to enhance the performance of the project development. The strategy of traditional concurrent engineering is also expected to be extended by forming a virtual concurrent engineering working environment through the exploration of the Internet-related application.

Bibliography

ABB (Asea Brown Boveri) 1992, 'CommTools User's Guide', ABB Robotics

Abdel-Wahab, Hussien M., Guan, Sheng-Uei, and Nievergelt, Jay 1988, 'Shared Workspaces for Group Collaboration: An Experiment Using Internet and UNIX Interprocess Communications', *IEEE Comm Magazine*,Vol. 26, No. 11, Nov, pp. 10-16

Abdel-Wahab, Hussien M. and Feit, Mark A. 1991, 'XTV: A Framework for Sharing X Window Clients in Remote Synchronous Collaboration', *Proc. of TRICOMM '91*, IEEE Conf. on Comm. Software: Communications for Distributed Applications and Systems, Chapel Hill, NC, USA, 18-19 April, pp. 159-167

ACM 1986, *Proc. CSCW'86*, Austin, Texas, USA., 1986

ACM 1988, *Proc. CSCW'88*, Portland, Oregon, USA., 1988

ACM 1990, *Proc. CSCW'90*, LosAngeles, California, USA., 1990

ACM 1992, *Proc. CSCW'92*, Toronto, Canada, USA., 1992

Altan, T., Oh, S.I., and Gegel, H.L. 1983, *Metal Forming: Fundamentals and Applications*, American Society for Metals

Autodesk 1996a, *Autodesk and Netscape to Extend Internet As Platform for Design and Engineering Collaboration*, Press Release, April [Online, accessed 26 March 1997] URL: http://www.autodesk.com/compinfo/pressrel/pr4_10f.htm

Autodesk 1996b, *Autodesk and Netscape Join to Make Online Design a Reality*, Press Release, October [Online, accessed 26 March 1997] URL: http://www.autodesk.com/compinfo/pressrel/pr10_31a.htm

Babadi, Aliasghar 1993, 'COMIX: A Tool to Share X Applications', in *Proceedings. of Second Workshop on Enabling Technologies Infrastructure for Collaborative Enterprises*, April, pp. 192-6

Baecker, Ronald M. 1993, *Readings in Groupware and Computer-Supported Cooperative Works: Assisting Human-Human Collaboration*, Morgan Kaufmann Publishers, Inc. ISBN 1-55860-241-0

Barron, Thomas 1996, Frame Relay at the U, [Online, accessed 19 March 1998] May 1996, Vol. 1, No. 2, Information Technology Newsletter, URL: http://www.umn.edu/oit/newsletter/0596-itn/frame-relay.html

Bell, Gavin, Parisi, Anthony and Pesce, Mark 1995, *The Virtual Reality Modelling Language Version 1.0 Specification* [Online, accessed 21 Oct. 1996] URL: http://vag.vrml.org/www-vrml/vrml.tech/vrml10-3.html

Bloom, Howard M. and Christopher, Neil 1997, 'A Framework for Distributed and Virtual Discrete Part Manufacturing', [Online, accessed 17 Jan. 1997], URL:

http://www.nist.gov/msidlibrary/doc/bloom96.htm, National Institute of Standards and Technology

Browsers 1996 [Online, accessed 19 Feb. 1996]
URL: http://www.rucc.net.au/tucows/softweb.html

Callaghan, Brent 1996, WebNFS: The Filesystem for the World Wide Web, SunSoft, Inc. May 1996 [Online, accessed 28 Oct. 1996]
URL: http://www.sun.com/solaris/networking/webfs/webfs.html

Cascade 1996, Frame Relay to ATM Interworking - Creating the Natural Flow to ATM, Management Briefing, CASCADE Communication Corp. April 1996, http://www.ascend.com/docs/datasheets/cascade/Framerelay.pdf

Chen, Liang-Chia 1990, CAD-Assisted In-Cycle Gauging System, M Engg. Thesis, Department of Mechanical Engineering, National Taiwan University, Taiwan, R.O.C.

Chen, Mon-Song, Shae, Zon-Yin, Kandlur, Dilip D., Barzilai, Tsipora P., and Vin, Harrick M. 1992, 'A Multimedia Desktop Collaboration System', in *Proceedings of GLOBECOM '92*, Comm. for Global Users, IEEE Global Telecommunications Conference, 1992, pp. 739-746

Chen, Yuh-Min and Hsiao, Yun-Tau 1997, 'A collaborative data management framework for concurrent product and process development', in the *Journal of Computer Integrated Manufacturing Systems*, Vol. 10, No. 4, pp. 446-469,

Choi, B.K., Chung, Y.C., Park, J.W. and Kim, D.H. 1994, 'Unified CAM-system architecture for die and mould manufacturing', *Computer-Aided Design*, Vol. 26, No. 3, March 1994, pp. 235-243

Comer, Douglas E. 1991, '*Internetworking with TCP/IP Volume I: Principles, Protocols and Architecture*', 2nd Edition, Prentice-Hall International Edition, Chapter 1

Connolly, John H. And Edmonds, Ernest A. (Eds.) 1994, *CSCW and Artificial Intelligence*, Springer-Verlag London Limited, ISBN 3-540-19816-4

Cousins, Robert E. 1992, 'Some Facts about NFS Response Time', AIM Technology, June 18, 1992

Dataquest 1995, 'Dataquest Perspective: NFS and Heterogeneous File Sharing - The Choice Is Easy', Client/Server Software Worldwide, Product Code: CSSW-WW-CR-9500, Publication Date: Agust 7, 1995

Davison, John M. 1988, *An Introduction to TCP/IP*, Spring-Verlag New York Inc., ISBN 0-387-96651-X, pp. 1-6

Deitz, Dan 1995, 'An infrastructure for integration', *Mechanical Engineering*, March 1995, pp. 78-80

Dowlatshahi, Shad 1994, 'A Comparison of Approaches to Concurrent Engineering', *The International Journal of Advanced Manufacturing Technology*, Vol. 9, pp. 106-113

Duffy, Alex H B 1995, 'Ensuring Competitive Advantage with Design Coordination', 2nd International Conference on Design to Manufacture in Modern Industry (DMMI'95), Bled, Solvenia, 29-30 May 1995, URL: http://www.cad.strath.ac.uk/publications/DMMI95.html

Ellis, Clarence A., Gibbs, Simon J., and Rein, Gail L. 1988, 'Groupware: The Research and Development Issues', MCC Technical Report No. STP-414-88, Software Technology Program, December

Ellis, C.A., Gibbs, S.J. and Rein, G.L. 1991, 'Groupware: Some Issues and Experiences', Communications of the ACM, January, Vol. 34, No. 1, pp. 39-58

Eltoweissy, Mohamed Youssef 1993, *A Framework for Data Sharing in Computer Supported Cooperative Environments*, PhD Dissertation, Computer Science, Old Dominion University, August

Erhorn, Craig and Stark, John, 1994, Competing by Design, Oliver Wight Publications, Inc., ISBN 0-939246-44-9, Chapter 6

Fanuc 1991a, *FANUC DNC 2 descriptions*, B-61992E/01, FANUC Ltd.

Fanuc 1991b, *Operator's manual: 0MF*, FANUC Ltd.

Faux, I.D. and Pratt M.J. 1979, *Computational geometry for design and manufacture*, Chichester, Eng : New York : Horwood ; Halsted Press, 1979, Series: Ellis Horwood series in mathematics and its applications, ISBN 047026473X (Halsted Press), Chap. 5

Feik, Dominic 1993a, 'Computer Supported Cooperative Work: A Literature Survey', Working Paper No. 3, The Union Research Centre on Office Technology (URCOT) July, ISSN 1320-2537

Feik, Dominic 1993b, 'Computer Supported Cooperative Work: A Literature Survey', Technical Report 93/9, Department of Computer Science, The University of Melbourne

Finin, Tim, Weber, Jay, Wiederhold, Gio, Genesereth, Michael, Fritzson, Richard, McGuir, James, Shapiro, Stuart, and Beck, Chris 1994, *Specification of the KQML Agent-Communication Language*, The DARPA Knowledge Sharing Initiative External Interfaces Working Group, DRAFT, February 9

Foo, S., Hui, S.C. and Daroy, C.J. 1996, 'MEMail - A Heterogeneous Multimedia Electronic Mail System', *International Journal of Information Technology*, Vol. 2, No. 1, pp. 55-77

Fraunhofer Center for Research in Computer Graphics 1993, *CSCW in Action* [Online, accessed 23 Oct. 1996],
URL: http://www.igd.fhg.de/www/idg-a9/projects/Siggraph93/siggraph93.html

Frost, Jim 1994, *BSD Sockets: A Quick and Dirty Primer* [Online, accessed 17 Oct. 1996], URL: http://ftp.std.com/homepages/jimf/sockets.html, Software Tool & Die, Copyright (c) 1988, 1994,

Gay, Robert K.L., Seet, P.K., and Lee, B.S. 1993, 'Network-based Concurrent Design Environment for Distributed-Based CAD', *Computing & Control Engineering Journal*, December 1993, pp. 253-260

Genesereth, Michael R. and Fikes, Richard E., et. Al. 1992, *Knowledge Interchange Format - Reference Manual*, Version 3.0, Computer Science Department, Stanford University, Stanford, California 94305

Geoview 1994, *Geoview: 3D Visualization Software* [Online, accessed 24 Oct. 1996] URL: http://www.geom.umn.edu/software/geomview/docs/oogltour.html

Goldman, James E. 1995, *Applied Data Communications - A business-oriented approach*, John Wiley & Sons, Inc., Chapter 9: Enterprise networking and client/server architectures, pp. 394-455

Goldstein, Cecil and Heard Ron 1991, 'Getting the most out of AARNet', Unix Version, Produced for Australian Academic and Research (Computer) Network by Computing Services, Queensland University of Techology, Nov., pp. 44-45

Gosling, James, Joy, Bill and Steele, Guy 1996, 'The Java Language Specification', Addison-Wesley, ISBN 0-201-63451-1

Gray, Matthew 1996, *Internet Growth Summary* [Online, accessed 12 Aug. 1996]. URL: http://www.mit.edu/people/mkgray/net/internet-growth-summary.html

Greenberg, Saul 1991, 'Computer-supported cooperative work and groupware: an introduction to the special issues', *Int. J. Man-Machine Studies*, Vol. 34, pp. 133-141

Greif, Irene, and Sarin, Sunil 1986, 'Data Sharing in Group Work', *in Proceedings of the Conference on Computer Supported Cooperative Work*, Austin, Texas

GRIP 1991, *GRIP Programming*, Volume 1 and Volume 2, McDonnell Douglas Corporation

Groover, Mikell P. 1987, *Automation, Production Systems, and Computer Integrated Manufacturing*, Prentice-Hall, Inc., ISBN 0-13-054610-0, Chap. 8-10

Grudin, Jonathan and Clement, Andrew 1994, 'Groupware and Social Dynamics: Eight Challenges for Developers', Communications of the ACM, January, Vol. 37, No. 1, pp. 92-105

Haase Bruce 1994, 'NEON: A Shinning Example of Design Efficiency', *Computer Graphics World*, February, pp. 41-44

Hao, Ming C. And Sventek, Joseph S. 1996, 'Collaborative design using your favorite 3D applications', Hewlett-Packard Research Labs, Palo Alto, CA, *in Proceedings of the Third ISPE International Conference on Concurrent Engineering*, Toronto, Ontario, Canada, August 26-28, pp. 8-15

Hart, Michael S. 1995, *Ask Dr. Internet* [Online, accessed 12 Aug. 1996]. URL: http://world.std.com/references/askdr/13may95

Hartley, John R. 1992, *Concurrent Engineering - Shortening Lead Times, Raising Quality, and Lowering Costs*, Productivity Press, Chapter 2

Hiltz, Starr Roxanne and Turoff, Murray 1978, *The Network Nation - Human Communication via Computer*, Addison-Wesley Pub. Co., Advanced Book Program, 1978

Hofman, J.K. 1993, 'Agile Manufacturing in the Process Industries', *Journal A*: Vol. 33, No. 3, pp. 47-50

Hsu, H.Y., Lin, Grier C.I and Kao, Y.C. 1997, 'Collaborative Planning on Generating Preliminary Assembly Sequence', Accepted for publication by the Second International ICSC Symposium on Intelligent Industrial Automation, September 17-19, Nîmes, France, pp. 7-13

Ishida, Haruhisa and Landweber, Lawrence H. 1993, 'Internetworking', Communications of the ACM, August, Vol. 36, No. 8, pp. 28-30

Ishii, Hiroshi, Kobayashi, Minoru, and Grudin, Jonathan 1993, 'Integration of Interpersonal Space and Shared Workspace: ClearBoard Design and Experiments', *ACM Transactions on Information Systems*, Vol. 11, No. 4, October, pp. 349-375

ISO 1996, *The Virtual Reality Modelling Language Specification - Version 2.0*, ISO/IEC CD 14772, August 4 [Online, accessed 21 Oct 1996],
URL: http://webspace.sgi.com/moving-worlds/spec

JATLite, JAVA Agent Template, Lite, 1997 URL: http://java.stanford.edu, Center for Design Research, Stanford University,

Jasnoch, Uwe, Kress, Holger, Schroeder, Klara, and Ungerer, Max 1994, *CoConut: Computer Support for Concurrent Design Using STEP*, in Proceedings of the Third IEEE Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises (WET-ICE) April 17-19, Morgantown, West Virginia, IEEE Computer Society Press [Online, accessed 23 Oct 1996] URL: http://www.igd.fhg.de/www/igd-a2/docs/pubs/pubs_1994/wetice/wetice_fullpaper.fm.html

Juhasz, Robert 1994, *XFS32 Version 1.0 Network File System (NFS) Client* (Windows version) [Computer program]. September 21, 1994 Copyright (C)

Kang, Suk-Ho, Kim, Namkug, Kim, Cheol-Young, Kim Yeongho and O'Grady Peter 1997, 'Collaborative Design using the World Wide Web', Technical Report TR97-02, IOWA Internet Laboratory, Department of Industrial Engineering, University of IOWA, USA. [Online, Accessed 24 Jan. 1997]
URL: http://128.255.23.188/Internetlab/Techrep/tr9702.pdf

Kao, Y.C. 1988, *Computer Aided Design and Manufacture on Extrusion Die Surface of Straight-Sided Spline*, (in Chinese) Master Thesis, Department of Mechanical Engineering, National Cheng Kung University, Tainan, Taiwan, R.O.C.

Kao, Y.C. and Lin, Grier C.I. 1995a, 'Integration of CAD/CAE/CAM on Metal Extrusion Die', in *Proceedings of The 6th International Conference on Manufacturing Engineering*, Melbourne, Australia, ICME95, 29 November - 1 December, pp. 523-528

Kao, Y.C. and Lin, Grier C.I. 1995b, 'A Case Study for International Remote Machining', in *Proceedings of The International Conference on Intelligent Manufacturing'95* (ICIM'95), June 14-17, Wuhan, China, pp. 556-563

Kao, Y.C. and Lin, Grier C.I. 1996a, 'Extending Single-location CAD/CAM for Multi-location Collaboration', in *Proceedings of the Third ISPE International Conference on Concurrent Engineering*, Toronto, Ontario, Canada, August 26-28, pp. 40-47

Kao, Y.C. and Lin, Grier C.I. 1996b, 'Implementation of a Collaborative CAD/CAM System', in *Proceedings of the The Fourth International Conference on Automation Technology*, Automation'96, Vol. 1, Mechanical Industry Research Laboratories/ITRI, Hsinchu, Taiwan, R.O.C., July 8-11, pp. 163-169

Kao, Y.C., Lin, Grier C.I. and He, Juan R. 1996c, 'Distributed Application of Die Surface CAD/CAM for Forward Metal Extrusion', in *Proceedings of the International Workshop on Automotive Manufacturing Science and Technology*, Jiao Tong University, Shanghai, China, 20-21 June, pp. 367-372

Kao, Y.C. and Lin, Grier C.I. 1996d, 'CAD/CAM Collaboration and Remote Machining', in the *Journal of Computer Integrated Manufacturing Systems*, Vol. 9, No. 3, pp. 149-160

Kao, Y.C. and Lin, Grier C.I. 1997a, 'Accessing a Remote Machining Cell through the World Wide Web', in Proceedings of the CIRP International Symposium -- Advanced Design and Manufacture in the Global Manufacturing Era -- August 21-22, 1997, Hong Kong, The University of Hong Kong, pp. 1-7

Kao, Y.C. and Lin, Grier C.I. 1997b, 'Development of a Collaborative CAD/CAM System', Accepted for publication by the Journal of Robotics and Computer-Integrated Manufacturing

Kennedy, E. Lee 1986, *CAD: design, drawing, data management*, Chapter 1, Whitney Library of Design, ISBN 0-8230-7104-9

Khanna, Sandeep, Sebrée, Michael, and Zolnowsky, John 1995, 'Realtime Scheduling in SunOS 5.0', SunSoft Incorporated, URL: http://computing.soongsil.ac.kr/~jmchoi/Realtime-Scheduling-in.html

Kim, Chang-Bong and Yang, Min-Yang 1992, 'Tool Path Verification and NC Program Editing Under a Multi-Window Environment', in *JAPAN/USA Symposium on Flexible Automation*, Vol. 1, ASME, pp. 409-414

Kim, Namkug, Kim Cheol-Young, Kang Suk-Ho, Kim, Yeongho and O'Grady Peter 1997, 'Interactive 3-D Dynamic Browsing of STEP Geometry Data Using the Internet and World Wide Web', Technical Report TR97-04, IOWA Internet Laboratory, Department of Industrial Engineering, University of IOWA, USA. [Online, Accessed 24 Jan. 1997] URL: http://128.255.23.188/Internetlab/Techrep/tr9704.pdf

Knister, Michael J. And Prakash, Atul 1990, 'DistEdit: A Distributed Toolkit for Supporting Multiple Group Editors', in *Proceedings of CSCW*, October, pp. 343-355

Kovács, George L. and Mezgár, István 1992, 'Concurrent Engineering: A Way to Design Intelligent Manufacturing Systems', JAPAN/USA Symposium on Flexible Automation, Vol. 1, pp. 461-466

Kress, Holger 1995, 'Integration Aspects within a Virtual Prototyping Environment', in Proceedings of the 5th International Conference on Flexible Automation and Intelligent Manufacturing - FAIM '95, June 28-30, Stuttgart, Germany [Online, accessed 22 Oct. 1996], URL: http://www.igd.fhg.de/www/igd-a2/projects/coconut_e.html

Kress, Holger 1996, *Shared 3D Viewer: Collaboration Support for CAD Users* [Online, accessed 22 Oct. 1996],
URL: http://www.igd.fhg.de/www/igd-a2/applications/3dviewer_e.html

Krol, E. And Hoffman, E. 1993, *FYI on "What is the Internet?"* [Online, accessed 6 Nov. 1996] ftp://nic.merit.edu/documents/fyi/fyi20.html

Leadwell 1993, *Electrical Manual for Vertical Machining Center with FANUC-0MF Controller - 1000CR*, Version: M9312-02/S4, Chapter 12, LEADWELL CNC Machines Manufacturing Corporation

Lee, J. 1995, 'Perspective and the world view on the integration of electro-mechanics in industry and education', (in Chinese) *J. of the Mechatronic Industry*, March 1995, No. 144, pp. 115-117, also Available Gopher: gopher://itrinews.itri.org.tw:70/00/tech/d1/d104a/9503/144-we

Levy, S., Subrahmanian, E., Konda, S. L., Coyne, R. F., Westerberg, A. W., and Reich, Y. 1993, 'An overview of the n-dim environment', Technical Report EDRC-05-65-93, Engineering Design Research Center, Carnegie Mellon University, Pittsburgh, PA.

Lin, Grier C. I. 1993, *Management of Advanced Manufacturing Technology*, Manufacturing Management Technology Seminar Lecture Notes, National Cheng Kung University, Tainan, Taiwan, July

Lin, Grier C.I, Kao, Y.C. and Lee, R.S. 1994, 'On the Real Time International Collaboration of CAD/CAM', in *Proceedings of The Third International Conference on Automation Technology*, July, Proceedings of Automation'94, Taipei, Taiwan, R.O.C., Vol. 1, pp. 281-284

Lin, Grier C.I., Kao, Y.C., Liaw, H.C., Lee, R.S., Chen, L.S, and Zhang, D. 1995a, 'Application of an Environment for International Collaborative CAD/CAM', ICCIM/AUTOFACT-ASIA-'95, in *Proceedings of The 3rd International Conference on Computer Integrated Manufacturing* in conjunction with the AUTOFACT-ASIA Exhibition, Singapore, pp. 323-330

Lin, Grier C.I. and Kao, Y.C. 1995b, 'Development of International Collaborative CAD/CAM', in *Proceedings of the Joint Australia-Korea Workshop on Manufacturing Technology*, 20-21 November, pp. 155-168

Lin, Grier C. I and Lu, T. F. 1995c, 'CAD-Based Intelligent Robot Workcell', in *Proceedings of the International Conference on Computer Integrated Manufacturing*, 11-14 July, Singapore, pp. 437-444

Lin, Grier C.I. and Hsu, Hung-Yao 1996, 'Preliminary Assembly Sequence Generation based on the Choice of Base Part', in *Proceedings of The Fourth International Conference on Automation Technology - Automation for Tomorrow's Competitiveness*, Mechanical Industry Research Laboratories/ITRI, Hsinchu, Taiwan, R.O.C., July 8-11, Vol. one, pp 347-353

Lottor, Mark and Rutkowski, Tony 1996 [Online, accessed 23 Oct 1996] URL: ftp://nic.merit.edu/statistics/nsfnet/history.hosts

Lu, Tien-Fu and Lin, Grier C. I. 1996, 'CAD, vision, and sensor based intelligent robot server', *Journal of Computer Integrated Manufacturing Systems*, Vol. 9, No. 2, pp. 91-100

Lutz, Robert A. 1994, 'Implementing Technologica; Change With Cross-Functional Teams', *Research Technology Management*, March - April, Vol. 37, No. 2, pp. 14-18

Malone, Thomas W. and Crowston, Kevin 1990, 'What is Coordination Theory and How Can It Help Design Cooperative Work Systems?', Center for Coordination Science, Massachusetts Institute of Technology, Cambridge, MA 02139, Proceedings of CSCW 90, October 1990

Marca, David and Bock Geoffrey 1992, *GROUPWARE: Software for Computer-Supported Cooperative Work*, Published by the IEEE Computer Society Press, ISBN 0-8186-2637-2, Chapter 1, 2, 7 and 8

McCullough, Malcolm 1995, 'Digital Design Media: Online Textbook', Chapter 1 The Second Industrial Revolution, Chapter 19 Virtual Design Studio, Chapter 20 What was computer-aided design?, [Online, accessed 15 Feb. 1996], URL: http://www.gsd.harvard.edu/~malcolm/DDM/DDMmain.html

McMahon, Chris and Browne, Jimmie 1993, *CAD/CAM from Principles to Practice*, Appendix C, Addison-Wesley publishing company, ISBN 0-201-56502-1

MCS 1993a, *Reference manual of GRAPL-IV*, Manufacturing and Consulting Services Inc.

MCS 1993b, *Reference manual of Extended GRAPL-IV*, Manufacturing and Consulting Services Inc.

MCS 1993c, *ANVIL-5000 SUN SYSTEM SUPPORT*, Manufacturing and Consulting Services Inc., Chapter 3

Michelis, Giorgio, Simone, Carla and Schmidt Kjeld 1993, Proceedings of the Third European Conference on Computer-Supported Cooperative Work - ECSCW'93, Kluwer Academic Publishers

Miller, Jeffrey G., De Meyer, Arnoud, and Nakane, Jinichiro, 1992, 'Benchmarking Global Manufacturing - Understanding International Suppliers, Customers, and Competitors', Richard D. IRWIN, Inc., ISBN 1-55623-674-3, Chapter 1

Newman, S.T., Rahimifard, S., and DE Souza R.B.R. 1995, 'Information management for the design of flexible machining cells', *Internattional Journal of Information Technology*, Vol. 1, No. 2, 1995, pp. 159-172

Newton, P.W., Wilson, B.G., Crawford, J.R., Clarke, S., and Mitchell, J. 1993, 'CAD Conferencing: Demonstration of Real-Time Wide Area Networking on a Design Project', in *Proceedings of a National Conference on New Applications for High Speed Metropolitan and Wide Area Networks in Construction and Manufacturing* at Telecom Research Laboratories, Melbourne on 29 July, CSIRO Division of Building, Construction & Engineering, Melbourne

O'Connor, Leo 1994, 'Agile manufacturing in a responsive factory', *Mechanical Engineering*, July, pp. 54-57

O'Hara-Devereaux, Mary and Johansen, Robert 1994, *Globalwork: Bridging Distance, Culture, and Time*, Jossey-Bass, Inc., Publisher, ISBN 1-55542-602-6, pp. 74-101

OpenGL 1998, URL: http://www.sgi.com/Technology/OpenGL, [Online, accessed 4 Feb 1998], Silicon Graphics Computer Systems

Park, Hisup, Tenenbaum, Jay M. And Dove, Rick 1993, Agile Infrastructure for Manufacturing Systems (AIMS) - A Vision for Transforming the US Manufacturing Base [Online, accessed 8 April 1995]
URL: http://www.eit.com/papers/DMC93/DMC93.html

Perreault, David John 1991, *Interactive Computer Support for Remote Design Teams: A New Approach*, Master Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, August 9

Peters, Ralph 1995, CrystalPad A Cooperative Annotation Environment [Online, accessed 23 Oct 1996] URL: http://www.igd.fhg.de/www/igd-a9/projects/CSCW-Tools/CSCW-Tools.html

Pferd, W., Peralta, L.A., and Prendergast, F.X. 1979, 'Interactive Graphics Teleconferencing', *IEEE Computer*, Vol. 12, No. 11, November, pp. 62-72

Press, Larry 1993, 'Before the Altair: The History of Personal Computing', *Communications of the ACM*, September, Vol. 36, No. 9, pp. 27-33

Puri, S 1996, 'The Ompact of Globalization on American Industry', Proceedings of the twelfth International Conference on CAD/CAM Robotics and Factories of the Future, Middlesex University, London, England, 14-16 August 1996, pp.466-470

Qikdraw Systems Pty. Ltd. 1993, *ASM Reference Manual*

Raulefs, Peter 1994, 'The Virtual Factory', in the Proceedings of the 13th World Computer Congress 94, Vol. 2, pp. 18-38, K. Brunnstein and E. Raubold (Editors), Elsevier Science B. V. (North-Holland), 1994 IFIP

Reich, Y., Konda, S., Monarch, I., and Subrahmanian, E. 1992, 'Participation and Design: An Extended View', In PDC'92: Proceedings of the Participatory Design Conference,

(Cambridge, MA), Muller, M. J., Kuhn, S., and Meskill, J. A. (eds.), CPSR, Palo Alto, CA.

Rembold, U., Nnaji, B. O., and Storr, A. 1993, *Computer Integrated Manufacturing and Engineering*, Addison-Wesley, Chapter 1 and 7

Renishaw Metrology Ltd. 1987, *User's guide - FANUC OM inspection software*, H-2000-6014-05-A

Resnick, Paul 1993, 'Phone-Based CSCW: Tools and Trials', *ACM Transactions on Information Systems*, Vol. 11, No. 4, October, pp. 401-424

Rodden, Tom 1991, 'A Survey of CSCW Systems', *Interacting with Computers*, Vol. 3, No. 3, pp. 319-353

Rogers, A S 1994, 'An Introduction to Groupware and CSCW', *BT Technology Journal*, Vol. 12, No. 3, July, pp. 7-11

Roseman, Mark 1993, *Design of a Real-Time Groupware Toolkit*, Master Thesis, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada

Salwen, Peter 1996a, 'Engineers on the Web', *Engineering World*, August 1996, pp. 8-12

Salwen, Peter 1996b, 'Sticking with the Web',URL:http://www.salwen.com/artcl-www.htm

Samba, 1997, [Online, accessed 3 Aug. 1997], URL: http://samba.anu.edu.au/

Santifaller, M. 1991, *TCP/IP and NFS Internetworking in a UNIX Environment*, Addison-Wesley

Sarin, Sunil and Greif, Irene 1985, 'Computer-Based Real-Time Conferencing Systems', IEEE *Computer*, October, pp. 33-45

Sasse, Martina Angela, Bilting, Ulf, Schultz, Claus-Dieter, and Turletti, Thierry 1994, 'Remote Seminars through Multimedia Conferencing: Experiences from the MICE project', *in Proceedings of INET'94/JENC5*, Technical Report, Department of Computer Science, University College London, also Available FTP: ftp://cs.ucl.ac.uk/mice/publications

Sasse, Martina Angela and Bennett, Roy 1995, 'Multimedia Conferencing over the Internet - The MICE Project', Technical Report, Department of Computer Science, University College London, Available FTP: ftp://cs.ucl.ac.uk/mice/ publications

Schroeder, Klara and Kress, Holger 1993, 'Distributed Conferencing Tools for Product Design', in Proceedings of the IFIP TC5/WG5.10 Working Conference on Interfaces and Industrial Systems for Production and Engineering, Darmstadt, Germany, pp. 283-294

Schulman, Michael A. 1995, 'Collaborative Communication in 3D', in *Proceedings of Concurrent Engineering - A Global Perspective (CE95)*, pp. 185-190

SGI 1995, IRIS Annotator [Online, accessed 1996]
URL: http://www.sgi.com/Products/Annotator.html

SGI 1996, *Deployment of Collaborative Computing: Best Practices* [Online, accessed 25 Oct. 1996] URL: http://www.sgi.com/Products/collaborate/best.html

ShowMe 1996, Product Description [Online, accessed 28 Oct. 1996] URL: http://www.sun.com/products-n-solutions/sw/ShowMe/products/ShowMe_Overview.html

Shu, Lily Hwei-Li, 1992, *Groupware Experiences in Three-Dimensional Computer-Aided Design*, Master Thesis, Mechanical Engineering, Massachusetts Institute of Technology

Shu, L. and Flowers, W. 1994, 'Teledesign: Groupware User experiments in three-dimensional computer-aided design', *Collaborative Computing*, Vol. 1, pp. 1-14

Spurr, Kathy, Layzell, Paul, Jennison, Leslie, and Neil Richards 1994, *Computer Support for Co-operative Work*, John Wiley & Sons

Stevens, W.Richard 1994, *UNIX Network Programming*, Prentice Hall, ISBN 981-3026-53-7, Chapter 1, Chapter 2, Chapter 3 and Chapter 6

Stratiquest 1996, The Benefits of Collaborative Computing - An Independent Multi-Company Study [Online, accessed 24 Oct. 1996]
URL: http://www.sgi.com/Works/Benefits_rpt_black.pdf

Subrahmanian, Eswaran, Coyne, Robert, Konda, Suresh, Levy, Sean, Marrtin, Richard, Monarch, Ira, Reich, Yoram and Westerberg, Arthur, 'Support System for Different-Time Different-Place Collaboration for Concurrent Engineering', The Proceedings of the 2nd IEEE Workshop on Enabling Technologies Infrastructure for Concurrent Engineering (WET ICE), Morgantown, West Virginia, April 1993, IEEE Computer Society Press, pp. 187-191

Sun Microsystems Inc. 1990, *Network Programming Guide*, Part No.: 800-3850-10, Revision A of 27 March, Chapter 1, 10, 11

Sun ISDN 1994, *Sun ISDN-BRI/SBI ISDN Hardware Configuration Guide*, Sun Publication No. 800-6863, June, Sun Microsystems, Inc.

SunLink ISDN 1993a, *SunLink ISDN Software Configuration Guide*, Sun Publication No. 800-7557, Sun Microsystems, Inc.

SunLink ISDN 1993b, *SunLink ISDN Software Reference Manual*, Sun Publication No. 800-7558, Sun Microsystems, Inc.

Tang, John C. and Minneman, Scott L. 1991, 'VideoDraw: A Video Interface for Collaborative Drawing', *ACM Transactions on Information Systems*, Vol. 9 No. 2, April, pp. 170-184

Tattam, P. R. 1992, *Trumpet HOPCHECK*, (Windows version) [Computer program] Copyright (c) 1992 by P.R. Tattam

Toye, George, Cutkosky, Mark R., Leifer, Larry J., Tenenbaum, J. Marty and Glicksman, Jay 1994, 'SHARE: A Methodology and Environment for Collaborative Product Development,' http://cdr.stanford.edu/pub/CDR/Publications/Reports/Share.ps

Traceroute 1995, *TRACEROUTE 1.2*, Lawrence Berkeley National Laboratory, Network Research Group, (Unix version) [Computer program] traceroute@ee.lbl.gov, Available FTP: ftp://ftp.ee.lbl.gov/traceroute-*.tar.Z

Trewitt, Glenn 1990, *Local-Area Internetworks: Measurements and Analysis*, Network Systems Laboratory (NSL) Research Report RP-1, Digital Network Systems Laboratory, available from e-mail address: NSL-Techreports@pa.dec.com

Udupa, K.S.N. 1992, *Computer supported cooperative work in specifying software requirements*, Master thesis of computer science, Montreal, Quebec, Canada, December

Watson, Todd 1996, 'Networked-centric computing', *Software Quarterly*, IBM's Magazine of Software Technologies, Vol. 3, No. 1, pp. 21-26

Young, Peter 1996, *Virtual Reality Systems* [Online, accessed 21 Oct. 1996] URL: http://www.dur.ac.uk/~dcs3py/pages/work/Documents/VR-Survey/index.html

Zientara, Marguerite 1981, *The history of computing - A bibliographical portrait of the visionaries who shaped the destiny of the computer industry*, ComputerWorld Communications, Inc. 1981, 375 Cochituate Rd., Box 880, Framingham, Mass. 01701

Publications List

A list of papers which were generated as a result of the work presented in this thesis is as follows:

Journal papers:

1. **Kao, Y C** and Lin, G C I, "CAD/CAM Collaboration and Remote Machining", Journal of Computer Integrated Manufacturing Systems, Vol. 9, No. 3, pp. 149-160, 1996

2. **Kao, Y C** and Lin, G C I, "Development of a Collaborative CAD/CAM System", Journal of Robotics and Computer-Integrated Manufacturing, Volume 14, Issue 1, pp. 55-68, 1998

Invited papers:

1. Lin, G C I, **Kao, Y C** and Lee, R S, "On the Real Time International Collaboration of CAD/CAM", Proceedings of The Third International Conference on Automation Technology, July 1994, Proceedings of Automation'94, Taipei, Taiwan, R.O.C., Vol. 1, pp. 281-284

2. Lin, G C I and **Kao, Y C**, "Development of International Collaborative CAD/CAM", Proceedings of the Joint Australia-Korea Workshop on Manufacturing Technology, 20-21 November 1995, pp. 155-168

3. Lin, G C I and **Kao, Y C**, "COCADCAM: A Collaborative CAD/CAM System", Proceedings of The Australia-Pacific forum on Intelligent Processing and Manufacturing of Materials (IPMM'97), July 14-17, 1997, Gold Coast, Australia, Vol. 2, pp. 1136-1142

Refereed Conference papers:

1. Lin, G C I, Hsu, H Y and **Kao, Y C**, "A Prototype for Real Time International Collaboration of Product Design", Proceedings of The 2nd Penang Electronics Manufacturing and Supporting Industries Exhibition and Conference, PEMAS'94, May 16-19, 1994, Penang Malaysia, pp. 63-70

2. Lin, G C I, **Kao, Y C**, Liaw, H C, Lee, R S, Chen, L S and Zhang, D, "Application of an Environment for International Collaborative CAD/CAM", ICCIM/ AUTOFACT-ASIA-'95, Proceedings of The 3rd International Conference on Computer Integrated Manufacturing in conjunction with the AUTOFACT-ASIA Exhibition, Singapore, 1995, pp. 323-330

3. **Kao, Y C** and Lin, G C I, "A Case Study for International Remote Machining", Proceedings of The International Conference on Intelligent Manufacturing'95 (ICIM'95), June 14-17, 1995, Wuhan, China, pp. 556-563

4. **Kao, Y C** and Lin, G C I, "Integration of CAD/CAE/CAM on Metal Extrusion Die", Proceedings of The 6th International Conference on Manufacturing Engineering, Melbourne, Australia, ICME95, 29 Nov. - 1 Dec. 1995, pp. 523-528

5. **Kao, Y C**, Lin, G C I and He, J R, "Distributed Application of Die Surface CAD/CAM for Forward Metal Extrusion", Proceedings of the International Workshop on Automotive Manufacturing Science and Technology, Jiao Tong University, Shanghai, China, 20-21 June 1996, pp. 367-372

6. **Kao, Y C** and Lin, G C I, "Implementation of a Collaborative CAD/CAM System", Proceedings of the The Fourth International Conference on Automation Technology, Automation '96, Volume 1, Mechanical Industry Research Laboratories/ITRI, Hsinchu, Taiwan, R.O.C., July 8-11, 1996, pp. 163-169

7. **Kao, Y C** and Lin, G C I, "Extending Single-location CAD/CAM for Multi-location Collaboration", Proceedings of the Third ISPE International Conference on Concurrent Engineering (CE96), Toronto, Ontario, Canada, August 26-28, 1996, pp. 40-47

8. **Kao, Y C** and Lin, G C I, "Application of a Collaborative CAD on Interactive Remote Coediting", Proceeding of the International Conference on Manufacturing Automation (ICMA'97), Department of Mechanical Engineering, The University of Hong Kong, April 28, 29 & 30, 1997, pp. 44-49

9. **Kao, Y C** and Lin, G C I, "Development of a Remote CAD/CAM Collaboration Framework", Proceedings of The 1st International Conference on Engineering Design and Automation, (EDA'97), The Central Plaza Hotel, Bangkok, THAILAND, March 18 - 21, 1997, pp. 667-670

10. **Kao, Y C** and Lin, G C I, "Accessing a Remote Machining Cell through the World Wide Web", Proceedings of The CIRP International Symposium -- Advanced Design and Manufacture in the Global Manufacturing Era -- August 21-22, 1997, Hong Kong, The University of Hong Kong, pp. 1-7

11. Hsu, H Y, **Kao, Y C** and Lin, G C I, "Collaborative Planning on Generating Preliminary Assembly Sequence", Proceedings of the Second International ICSC Symposium on Intelligent Industrial Automation, September 17-19, 1997, Nîmes, France, pp. 7-13

12. **Kao, Y C** and Lin, G C I, "A Collaborative System for CAD-Geometry Coediting and Planning", Proceedings of the 4th International Conference on Manufacturing Technology, Hong Kong, 30 Nov ~ 3 Dec 1997, pp. (in a CD, no page information)