TELECONTROL

This article presents remote machining, that is, conducting the machining operations at a distance through "remote control." *Remote control* is defined in the *Oxford Encyclopedic English Dictionary* (1) as "control of a machine or apparatus from a distance by means of signals transmitted . . .". Thus remote machining is defined in this article as "to conduct or perform a machining task at geographically separated locations through the transmission of electrical signals."

It is not uncommon for a company to have multiple facilities installed at separated geographical locations; for example, in a multinational enterprise. The separation of the available facilities hinders communication within a company due to possible time and language constraints (2). Very often, in the manufacturing domain, the investment of machine tools (e.g., the cost of a five-axis machining center) is very expensive. To operate expensive equipment not only locally but remotely would increase operational efficiency and economy of the investment (3).

Advances in computer and communications technologies have made computer-assisted operations potentially the most convenient and powerful mechanism for geographically distributed groups of people wishing to communicate (4,5). The Internet came into existence as a natural development which was a result of the Advanced Research Projects Agency's network (ARPANET) and the National Science Foundation's network (NSFNET) in the 1980s (6,7) providing services such as e-mail, ftp, telnet, facsimile, and the World Wide Web (WWW) (8,9), and so forth. The World Wide Web has been the most popular application of the Internet. This implies that worldwide devices can be operated remotely if they are linked properly to form a networked computer-integrated environment for distributed application (10,11). Examples include a remote machining cell (12) that can be accessed through the World Wide Web and an unstaffed factory (13) that can be maintained or troubleshot at a distance.

This article illustrates practices in implementing networked remote machining operations.

REMOTE MACHINING

This section uses two examples to explain the remote machining practices. One is an unstaffed factory and the other is a remote machining cell.

Unstaffed Factory

Network-integrated manufacturing system is the trend in computer-integrated manufacturing. Since the term computer-integrated manufacturing (CIM) was coined by Harrington (14) in 1973, its definition has constantly changed. Typical computer-integrated manufacturing activities include computer-aided design (CAD), computer-aided process planning (CAPP), computer-aided manufacturing (CAM), and production planning and control (PP&C) (15,16). A distinction about CIM was made by Savage (17) with the advent of network technology: CIM I and CIM II, the latter emphasizing the incorporation of network technology resulting in a network-integrated manufacturing system. Network technology is necessary in implementing a network-integrated manufacturing system.

> If a CIM II system is only accessible by remote users through network, the underlying manufacturing system becomes an unmanned factory (13).

A network-integrated manufacturing system constructed at the National Chiao-Tung University (13) (NCTU) is used as an example to demonstrate a remotely operable unstaffed factory through the Internet for machining operation at a distance. The network-integrated manufacturing system at NCTU is described in the following sections: remote user interface, network configuration, and control architecture.

Remote User Interface. A remote end user and the network-integrated unstaffed factory are linked through the Internet,

J. Webster (ed.), *Wiley Encyclopedia of Electrical and Electronics Engineering*. Copyright © 1999 John Wiley & Sons, Inc.

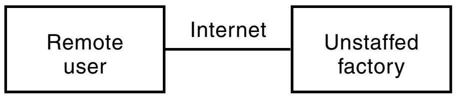

Figure 1. The linkage between a remote end user and the unstaffed factory.

as shown in Figure 1. The remote user's screen can display the graphic feedback showing the up-to-date movement of parts dynamically. Cameras were installed in the factory and a remote user can adjust the focus, zoom, and movement of the camera.

Network Configuration. The unstaffed factory is controlled through an Ethernet (10 Mbps) network. The network configuration of the unstaffed factory includes the campus Fiber Distributed Data Interface (FDDI) network (100 Mbps), T3 line (44.7 Mbps), and Asynchronous Transfer Mode (ATM) switch (200 Mbps) for long distance transmission. The longest distance that has been successfully tested is about 86 km and the data transmission went through the Ethernet, routers, the FDDI to the National Center for High-Performance Computing (NCHC), to an Ethernet-to-ATM switching hub (80 Mbps), ATM switch, T3 line, and to another ATM.

Control Architecture. There are four levels in the unstaffed factory: planning, execution, control, and manufacturing. At the manufacturing level, the physical manufacturing device (PMD) consists of computer numerical control (CNC) machine tools, robots, conveyors, automated guided vehicle (AGV), cameras, monitoring equipment, mold filling machine, and so forth. At the control level, the controller includes Allen Bradley programming logic circuits (PLCs) with I/O modules. At the execution level, the expert system software is G2 from Gensym (18). Other software that is used in the network-integrated manufacturing system includes: Oracle, Oracle-G2 Bridge, G2-Interface Bridge, Interchange, and ladder diagrams. Mathematical models such as Petri net theory (19), automata-based supervisory control theory, calculus of communication systems, and so forth were adopted for modeling the behavior of concurrent activities. Two more manufacturing systems, Simulated Manufacturing System (SMS) and Virtual Manufacturing System (VMS), were developed to control the system. The physical activity of the PMD is mapped temporarily by the simulated activity of SMS. Both PMD and the SMS act like data servers of the VMS which is a graphic computer interface. The VMS is used to control the system remotely.

Summary. This network-integrated manufacturing system has been successful in such a way that an unstaffed environment has been constructed and could be shared by students from other schools or cities if they were properly networked. The on-site closed, hot, humid, noisy, and uncomfortable environment has been replaced with an open, safe, and more comfortable remote manufacturing environment. Remote diagnosis has also been implemented for the recovery of machine breakdown. However, the movement of the robot and the machining task of the machining center were predetermined. For example, the system cannot accept an external numerical control program and conduct the machining task using existing workpieces.

Remote Machining Cell

This section describes a remote machining cell (RMC) that is accessible through the Internet (12). The strategies adopted to develop the RMC are presented and the configuration of the RMC is described. The result of this development is an RMC that can be accessed through the World Wide Web by geographically dispersed designers and manufacturing engineers in performing collaborative work for NC milling machining. The IP address mentioned in this section is only for reference. An overview of the remote machining cell is described first and a description of the system architecture in accessing the remote machining cell follows.

Overview of the RMC. The remote machining cell integrates a machining center for machining operation, and an intelligent robot for workpiece loading and unloading tasks. The configuration of the system is shown in Figure 2. The machining center was equipped with a Fanuc 0MF controller and was controlled by a personal computer (PCM) through the DNC2 interface (20), the robot was equipped with an ABB robot controller (21) controlled by another personal computer (PCR) through CommTools (22). The robot was also equipped with vision and force/torque sensor (23) capability in order to have the intelligence to automatically load and unload the requested workpiece to and from the table of the machining center. The operating system of the PCs was Windows for Workgroups 3.11. The communication protocols for both the machining center and the robot were the RS-232 protocols. There were three Sun workstations (WKs); one (*camr3*) ran a vision system for the robot to search for the correct workpiece; the other (*camr1*) ran the NCSA http (24) (the HyperText Transfer Protocol) server program for the remote machining cell to be accessed through the WWW. A collaborative CAD/CAM system, *COCADCAM* (25) was implemented in two workstations, *camr1* and *camr2*. The *COCADCAM* system provided an environment for distributed CAD/CAM users to concurrently coedit the CAD geometry and to generate an NC program for the machining operation to be executed in the remote machining cell. The physical locations of the workstations—*camr1, camr2,* and *camr3*—could be geographically dispersed if they were connected through the Internet.

Communication of the Controlling Programs in RMC. The communication between the controlling and monitoring programs of both the machining center and the robot was through the exchange of data files. These data files were kept in a common database repository located on a Sun workstation and were accessible through the network file system (PC–NFS) linking the PCs and the workstations. Although the sharing of the file system (hard disk resource) was a basic function between Unix systems and PCs, it was not the case between the operating systems of Unix and PC. (A Unix operating system could be installed on PC, but the personal computers described in this article did not run Unix. The operating system provided by the PC Windows environment could share the other computer's hard disk directory through

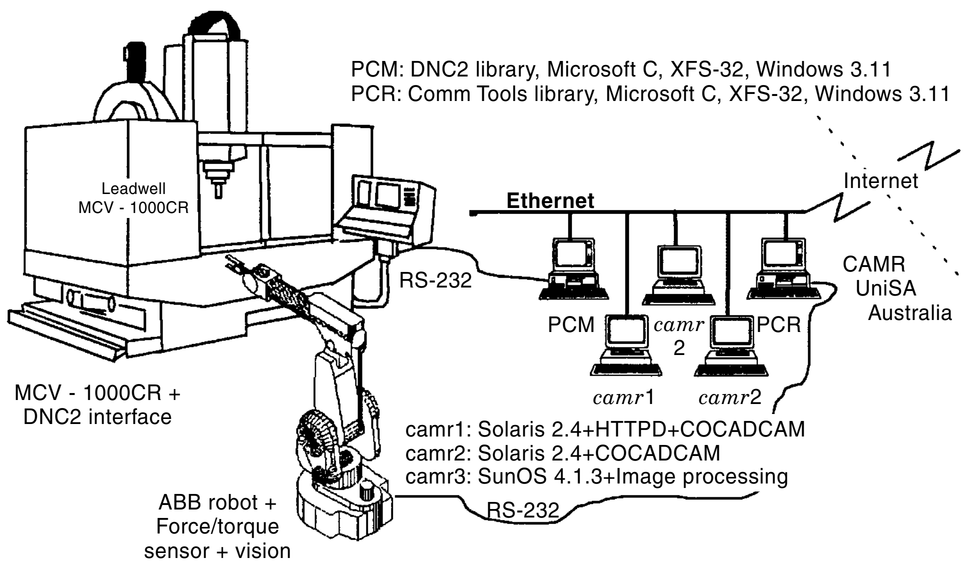

Figure 2. The system configuration of the remote machining cell.

the file manager and/or the explorer but not with Unix.) Therefore, to share the file system of the Sun workstation from a PC, a PC–NFS demon process (a program that is used as a server process to enable the sharing of the network file system) was installed on the Sun workstation and a client process XFS32 (26) was installed on the PC as well. The XFS32 was adopted to share the file system from the PC to a Sun workstation.

System Architecture in Accessing the RMC. The system architecture for accessing the RMC through the WWW is shown in Figure 3. The input to the RMC was through a data file and therefore a common gateway interface (CGI) (27) program was implemented to respond to requests for the RMC operation from the fill-out forms (28) of a WWW browser and to output command messages in a data file to the RMC. The RMC was then actuated if requested. The status of the RMC was updated regularly and stored in a data file in HyperText Modeling Language (HTML) (29) format. There were two open standards-based mechanisms available for the WWW client to view the dynamic documents which resided in the WWW server. These mechanisms (30) were called *server-push* and *client-pull.* In the *server-push* mechanism, a connection between the WWW client and server was made for an indefinite period of time until either the server sent the data and a terminator to the client or the client stopped the connection. In the *client-pull* mechanism, the connection was not kept but the client was able to make a new connection with and receive the document from the server after a certain period of time. The mechanism of *client-pull* rather than *server-push* was adopted to reduce the load of the server computer. The status file generated from the RMC was merged

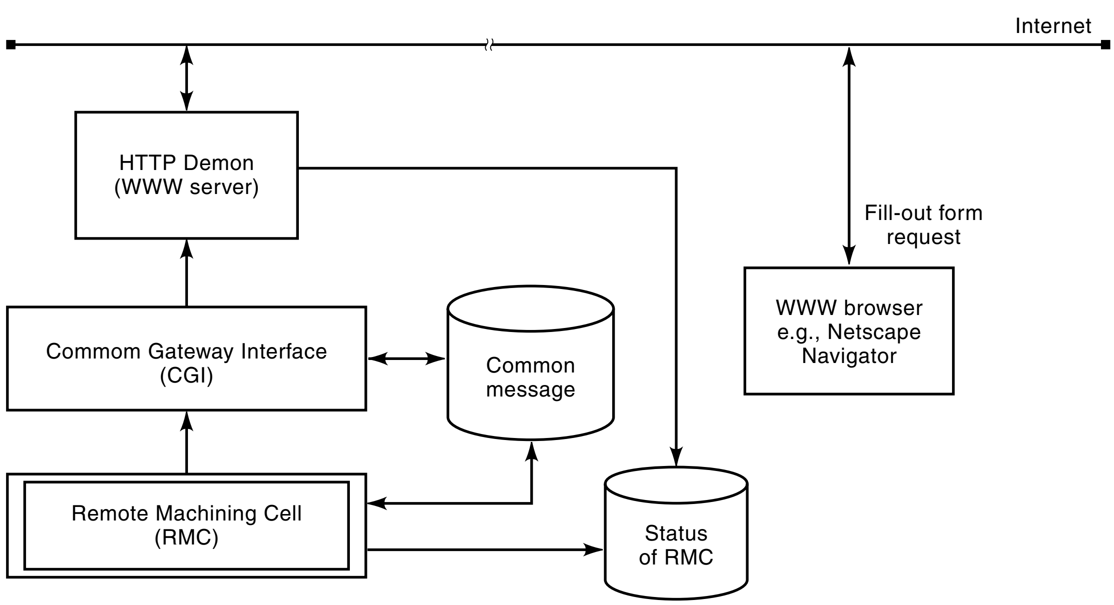

Figure 3. The system architecture in accessing the RMC.

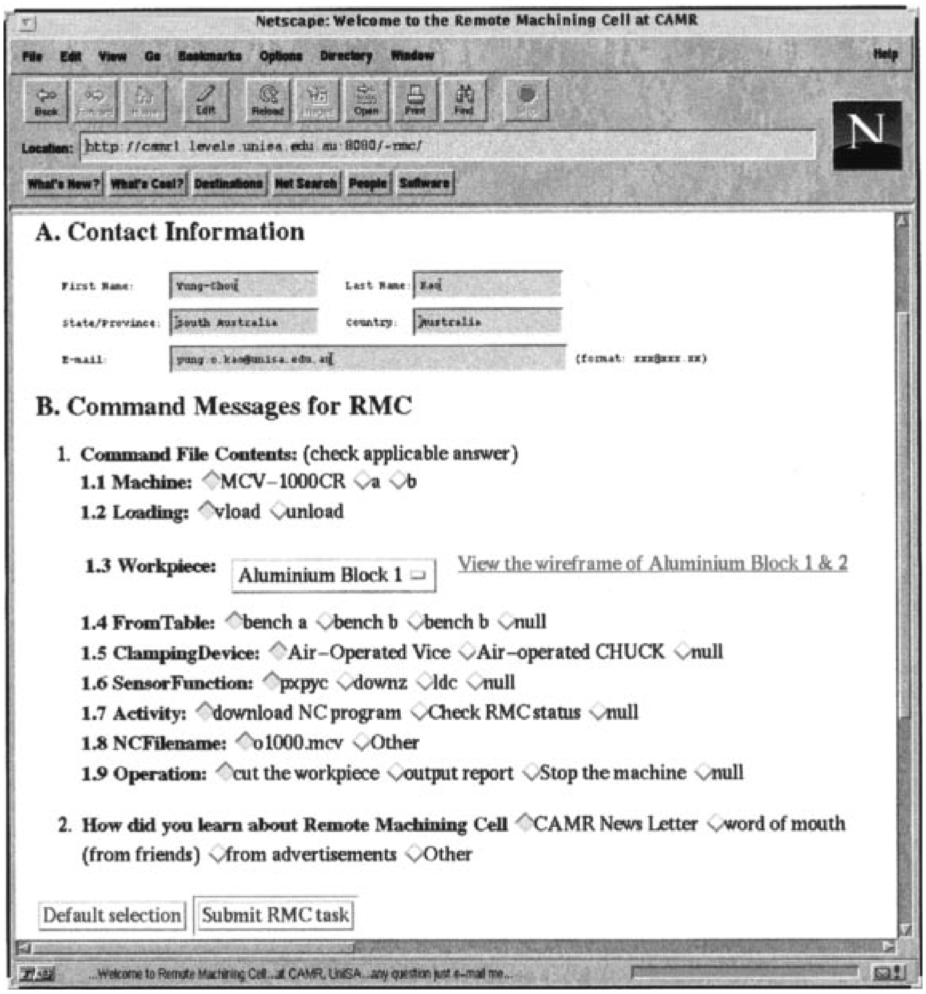

Figure 4. The WWW interface in accessing the RMC.

with the main status file document by applying the *server-side includes* (31) directive. Therefore, the status of the RMC could be dynamically updated by applying (1) the CGI program, (2) the *client-pull* mechanism and (3) the *server-side includes* directive. An appropriate audio file was also included in the RMC status file so that the status of the RMC could be audio transmitted to the user if the WWW browser supported Java (32) applet (application) and could access sound speakers.

Example of the RMC. This section illustrates how the RMC could be accessed through the WWW with Netscape Navigator (33) as the WWW browser. First, a WWW client opened a connection to the WWW server, as shown in Figure 3, accessing the HTML document http://camr1.levels.unisa.edu.au:8080/~rmc. The browser then displayed the fill-out form, as shown in Figure 4, and the HTML source codes as shown in Appendix 1, after receiving a response from the server. The client needed to fill in the contact information and the command messages before submitting a task for RMC operation. There were nine records to be assigned in the command file contents. The first record was the machine station (only one machining center, MCV-1000CR, was available). The second record was the method used for the robot to handle the workpiece, where there were two options: VLOAD (using vision system to assist loading tasks) and UNLOAD. The third record was the encoded name of the workpiece; for example, ALBLK1 represents an aluminium block 1 and ALBAR1 represents an aluminum bar 1. The fourth record assigned the workbench that stored the workpiece. The fifth record assigned the clamping device, for example, the air-operated vice. The sixth record concerned the force/torque sensor function for the robot to use in handling the workpiece. Records seven to nine assigned the operation of the RMC. If the seventh record was "download NC program," then the eighth record had to be "o1000.mcv" if this NC program was generated by the *COCADCAM* system. If the ninth record was "cut the workpiece" the RMC could be actuated for the machining operation. These records were assigned by the client interactively. This fill-out form was submitted by clicking the "Submit RMC task" button. Once this form was submitted the CGI program *cgi-rmc* was contacted through the WWW server to confirm the submission of the task, as shown in Figure 5 and the source code of the C program of cgi-rmc shown in Appendix 2, and a spoken greeting "welcome to the remote machining cell" would be transmitted and played through the Java applet. Also, the *client-pull* mechanism was actuated and used to update the status of the RMC, as shown in Figure 6.

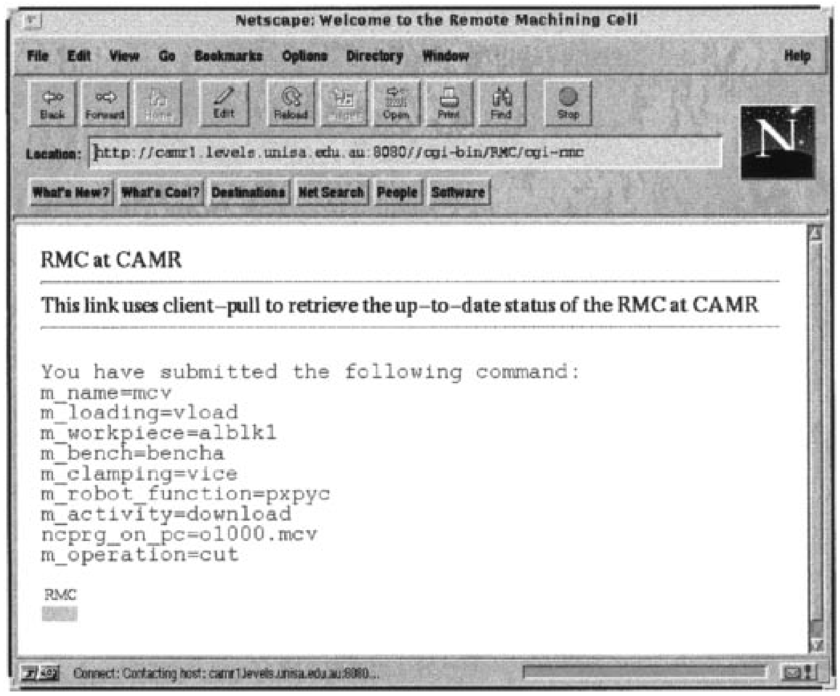

Figure 5. The response of the HTTP server CGI program *cgi-rmc*.

Summary. The algorithms in implementing a remote machining cell that can be accessed through the World Wide Web include the adoption of the common gateway interface, the client-pull mechanism, and the Java applet to show the status of a remote machining cell, by downloading text and audio information on demand, to a remote WWW client. The remote machining cell was constructed to perform the NC machining task following the work which resulted from a collaborative CAD/CAM system *COCADCAM* that output the NC program. The WWW browser could then be used to submit the machining task through the Internet for a remote machining operation conducted by a remote machining cell and to check the status of the remote machining cell.

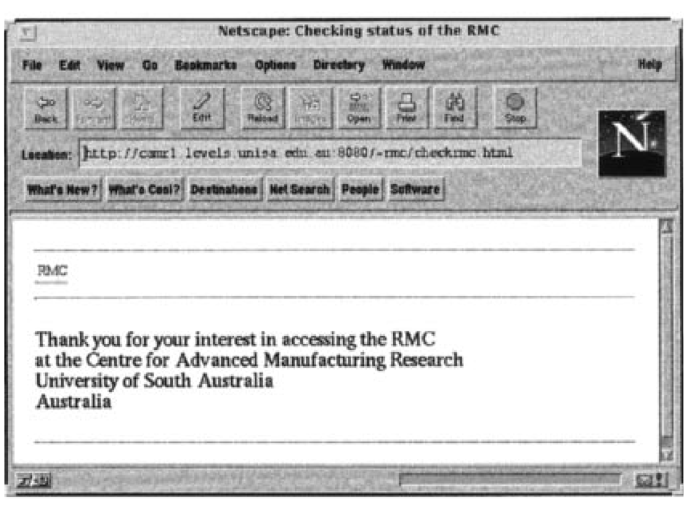

Figure 6. Updating the status of the RMC through *client-pull* mechanism.

EVALUATION AND FUTURE DEVELOPMENT

Distributed operations have emerged from advances in network and processor technology and the increase in multilocation enterprises. A remote manufacturing system such as a remote machining cell allows the implementation of a manufacturing system that becomes borderless, that is, to be accessed through the Internet without physical barrier. Its major benefits are: (1) cost reduction in such areas as transportation of raw material; (2) enabling facilities to integrate enterprisewide business functions and to achieve a higher degree of agility and responsiveness to market changes; (3) worldwide plug-and-play integration, merging remote machining into computer-integrated manufacturing system.

Development and implementation of remote machining depend on the availability of distributed, heterogeneous, multivendor computing platforms and communication networks with standardized interfaces and services such as the World Wide Web. Multimedia could be incorporated to enhance the control interface. Open architecture control (34) for machine tools should also be explored. A virtual factory (globally networked and integrated manufacturing facilities-based) environment (35) and diagnostic agents (36) can be applied to integrate the distributed remote machining cell.

APPENDIX 1. THE HTML CODES IN ACCESSING THE RMC

```
⟨HTML⟩
⟨HEAD⟩
⟨META HTTP-EQUIV="content-type" CONTENT="text/html; charset=big5"⟩
⟨title⟩Welcome to the Remote Machining Cell at CAMR⟨/title⟩
⟨HEAD⟩
⟨BODY onLoad="person_in(); timerONE=window.setTimeout('scrollit_r21(100)',500);"
onUnload='person_out()' BGCOLOR=#FFFFBF LINK=#0000ff VLINK=#ff0000 TEXT=#000000⟩⟨LEFT⟩
⟨hr⟩
⟨font size=3⟩Fill out this form completely to submit a COMMAND ⟨b⟩asking for Remote Machining
Cell⟨/b⟩ to operate. ⟨br⟩
After filling out this form, click the Submit RMC task button at the end of the page.⟨/font⟩⟨P⟩
⟨h3⟩A. Contact Information⟨/h3⟩
⟨form action="http://camr1.levels.unisa.edu.au:8080//cgi-bin/RMC/cgi-rmc" method=POST⟩
⟨pre⟩
  First Name:   ⟨input type=text size=20 name=First value=" "⟩     &Last Name:⟨input type=text size=20 name=Last-
  Name value=" "⟩
  State/Province: ⟨input type=text size=20 name=State value="South Australia"⟩     Country: ⟨input type=text size=20 na-
  me=Country value="Australia"⟩
  E-mail:     ⟨input type=text size=57 name=Email value=" "⟩ (format: xxx@xxx.xx)
⟨/pre⟩
⟨h3⟩B. Command Messages for RMC⟨/h3⟩
⟨ol⟩
⟨li⟩⟨b⟩Command File Contents:⟨/b⟩(check applicable answer)
⟨dl⟩
⟨dt⟩⟨b⟩1.1 Machine:⟨/b⟩
⟨input type=radio name=Machine value=mcv CHECKED⟩MCV-1000CR
⟨input type=radio name=Machine value=a⟩a
⟨input type=radio name=Machine value=b⟩b⟨br⟩
⟨dt⟩⟨b⟩ 1.2 Loading: ⟨/b⟩
⟨input type=radio name=Loading value=vload CHECKED⟩vload
⟨input type = radio name=Loading value=unload⟩unload
⟨br⟩
⟨TABLE CELLSPACING=1⟩
⟨TR⟩⟨TD VALIGN=center⟩⟨dt⟩⟨b⟩1.3 Workpiece:⟨/b⟩⟨/TD⟩
  ⟨TD VALIGN=top⟩
    ⟨SELECT name=Workpiece SIZE="1"⟩
    ⟨OPTION VALUE="alblk1" SELECTED⟩Aluminium Block 1
    ⟨OPTION VALUE="alblk2"⟩Aluminium Block 2
    ⟨OPTION VALUE="albar1"⟩Aluminium Bar 1
    ⟨OPTION VALUE="albar2"⟩Aluminium Bar 2
    ⟨OPTION VALUE="null"⟩null⟨br⟩
    ⟨/SELECT⟩
  ⟨/TD⟩
  ⟨TD VALIGN=center⟩⟨a
href="http://camr1.levels.unisa.edu.au:8080/yungchou/RMC/workpiece.html;"⟩
      View the wireframe of Aluminium Block 1 & 2⟨/a⟩
  ⟨/TD⟩
⟨/TR⟩
⟨/TABLE⟩

⟨dt⟩⟨b⟩1.4 FromTable: ⟨/b⟩
⟨input type=radio name=FromTable value=bencha CHECKED⟩bench a
⟨input type=radio name=FromTable value=benchb⟩bench b
⟨input type=radio name=FromTable value=benchc⟩bench b
⟨input type=radio name=FromTable value=null⟩null⟨br⟩

⟨dt⟩⟨b⟩1.5 ClampingDevice:⟨/b⟩
⟨input type=radio name=ClampingDevice value=vice CHECKED⟩Air-Operated Vice
⟨input type=radio name=ClampingDevice value=chuck⟩Air-operated CHUCK
⟨input type=radio name=ClampingDevice value =Other⟩null⟨br⟩
```

```
⟨dt⟩⟨b⟩1.6 SensorFunction:⟨/b⟩
⟨input type=radio name=SensorFunction value=pxpyc CHECKED⟩pxpyc
⟨input type=radio name=SensorFunction value=downz⟩downz
⟨input type=radio name=SensorFunction value=ldc⟩ldc
⟨input type=radio name=SensorFunction value=null⟩null⟨br⟩

⟨dt⟩⟨b⟩1.7 Activity:⟨/b⟩
⟨input type=radio name=Activity value=download CHECKED⟩download NC program
⟨input type=radio name=Activity value=status⟩Check RMC status
⟨input type=radio name=Activity value=null⟩null⟨br⟩

⟨dt⟩⟨b⟩1.8 NCFilename:⟨/b⟩
⟨input type=radio name=NCFilename value=o1000.mcv CHECKED⟩o1000.mcv
⟨input type=radio name=NCFilename value=Other⟩Other⟨br⟩

⟨dt⟩⟨b⟩1.9 Operation:⟨/b⟩
⟨input type=radio name=Operation value=cut CHECKED⟩cut the workpiece
⟨input type=radio name=Operation value=report⟩output report
⟨input type=radio name=Operation value=stop⟩Stop the machine
⟨input type=radio name=Operation value=null⟩null⟨/dl⟩⟨br⟩

⟨li⟩⟨b⟩How did you learn about Remote Machining Cell⟨/b⟩
⟨input type=radio name=LearnRMU value=CAMRNewsLetter CHECKED⟩CAMR News Letter
⟨input type=radio name=LearnRMU value=Words⟩word of mouth (from friends)
⟨input type=radio name=LearnRMU values=Ads⟩from advertisements
⟨input type=radio name=LearnRMU value=Other⟩Other
⟨p⟩
⟨/ol⟩

⟨!h3⟩⟨hide C. Make sure you already have an account to access the RMC⟩⟨!/h3⟩
⟨font color=green⟩ ⟨input type="reset" value="Default selection"⟩⟨/font⟩
⟨font color=blue⟩ ⟨input type="submit" value="Submit RMC task"⟩⟨/font⟩⟨p⟩
⟨/form⟩
⟨hr⟩
⟨/BODY⟩⟨/HTML⟩
```

APPENDIX 2. THE CGI CODES IN RESPONDING THE REQUEST OF THE RMC

```
/* filename: cgi-rmc.c */
#include ⟨stdio.h⟩
#ifndef NO−STDLIB−H
#include ⟨stdlib.h⟩
#else
char *getenv();
#endif
#include ⟨string.h⟩

#define MAX_ENTRIES 10000
#define MAXLEN 80

typedef struct {
   char *name;
   char *val;
} entry;

char *makeword(char *line, char stop);
char *fmakeword(FILE *f, char stop, int *len);
char x2c(char *what);
void unescape_url(char *url);
void plustospace(char *str);
```

```
char * not_empty;
char * is_empty = "⟨empty⟩";

main(int argc, char *argv[])
{
   entry entries[MAX_ENTRIES];
   register int x,m=0;
   int cl, flag_cut=0, m_job_number_val=0;
/* macro for displaying environment variables */
#define gotenv(a) ((not_empty = getenv(a)) ? not_empty:is_empty)
   char m_name[20], m_job_number[5], m_loading[20], m_workpiece[20];
   char m_bench[20], m_clamping[20], m_robot_function[20], m_activity[20], String_to_RMC[80];
unsigned char ncprg_pc[20];
char m_operation[20], m_status[20], ncprg_on_cnc[20], ch[MAXLEN], sp[2], out_name[MAXLEN];
FILE *fp;

printf("Content-type: text/html%c%c",10,10);
if (strcmp(getenv("REQUEST_METHOD"),"POST")) {
   printf("This script should be referenced with a METHOD of POST./n");
   printf("If you do not understand this, see ");
   printf("⟨A HREF=\"http://www.ncsa,uiuc.edu/SDG/Software/Mosaic/Docs/fill-out-forms/overview.html\"⟩forms overview⟨/
   A⟩.%c",10);
   exit(1);
  }
   if(strcmp(getenv("CONTENT_TYPE")," application/x-www-form-urlencoded")) {
   printf("This script can only be used to decode form results.");   exit(1);
  }
   cl = atoi(getenv("CONTENT_LENGTH"));
   for(x=0;cl && (!feof(stdin));x++) {
      m=x;
      entries[x].val = fmakeword(stdin,'&rsquo,&cl);
      plustospace(entries[x].val);
      unescape_url(entries[x].val);
      entries[x].name = makeword(entries[x].val,'=');
   }

   printf("⟨HTML⟩%c⟨HEAD⟩RMC at CAMR⟨/HEAD⟩%c",10,10);
   printf("⟨TITLE⟩Welcome to the Remote Machining Cell⟨/TITLE⟩");
   printf("⟨body bgcolor=\"#FFFFBF\"⟩"); printf("⟨\body⟩");

   for(x=0; x ⟨= m; x++) {
   if (!strcasecmp(entries[x].name, "Machine")) {
         strcpy(m_name, entries[x].val);      if ( !strcasecmp(m_name, "mcv") ) { }
      else
         {
         printf("⟨pre⟩The machine ⟨code⟩⟨font size =+6 COLOR=BLUE⟩ %s⟨/font⟩");
         printf("%s ⟨/font⟩ is not available yet⟨/code⟨=/pre⟩", m_name);   exit(1);
         }
}
else if (!strcasecmp(entries[x].name, "Loading")) {
         strcpy(m_loading, entries[x].val);
         if ( !strcasecmp(m_loading, "vload") || !strcasecmp(m_loading, "load") ) { }
         else
            {
            printf("⟨pre⟩The loading method ⟨code⟩⟨font size =+6 COLOR=BLUE⟩");
            printf("%s⟨/font⟩ is not available yet⟨/code⟩⟨/pre⟩", m_loading);   exit(1);
            }
}
else if (!strcasecmp(entries[x].name, "Workpiece")) {
         strcpy(m_workpiece, entries[x].val);
```

```
        if ( !strcasecmp(m_workpiece, “alblk1”) ) { }
        else
          {
          printf(“⟨pre⟩The workpiece ⟨code⟩⟨font size =+6 COLOR=BLUE⟩”);

          printf(“%s ⟨/font⟩ is not available yet⟨/code⟩⟨/pre⟩”, m_workpiece);   exit(1);
          }
}
else if (!strcasecmp(entries[x.name, “FromTable”)) {
        strcpy(m_bench, entries[x].val);
  if ( !strcasecmp(m_bench, “bencha”) ) { }
  else
    {
        printf(“⟨pre⟩The bench ⟨code⟩⟨font size =+6 COLOR=BLUE⟩ ”);
        printf(“%s ⟨/font⟩ is not available yet⟨/code⟩⟨/pre⟩”, m_bench);    exit(1);
        }
}
else if (!strcasecmp(entries[x].name, “ClampingDevice”)) {
        strcpy(m_clamping, entries[x].val);
        if ( !strcasecmp(m_clamping, “vice”) ) { }
        else
          {
          printf(“⟨pre⟩The clamping device ⟨code⟩⟨font size =+6 COLOR=BLUE⟩”);
          printf(“%s⟨/font⟩ is not available yet⟨/code⟩⟨/pre⟩”, m_clamping);   exit(1);
          }
  }
  else if (!strcasecmp(entries[x].name, “SensorFunction”)) {
        strcpy(m_robot_function, entries[x].val);
        if ( !strcasecmp(m_robot_function, “pxpyc”) ) { }
        else
          {
          printf(“⟨pre⟩The robot loading function ⟨code⟩⟨font size =+6 COLOR=BLUE⟩ ”);
          printf(“ %s ⟨/font⟩ is not available yet⟨/code⟩⟨/pre⟩”, m_robot_function);   exit(1);
          }
  }
  else if (!strcasecmp(entries[x].name, “Activity”)) {
        strcpy(m_activity, entries[x].val);
        if ( !strcasecmp(m_activity,“status”) ) {
}        else if ( !strcasecmp(m_activity,”download”) ) { }
        else
          {
          printf(“⟨pre⟩The Activity ⟨code⟩⟨font size =+6 COLOR=BLUE} %s⟨/font⟩”,m_activity);
          printf(“ is not available yet⟨/code⟩⟨/pre⟩”);
          exit(1);
          }
  }
  else if (!strcasecmp(entries[x].name, “NCFilename”)) {
        strcpy(ncprg_on_pc, entries[x].val);
        if ( !strcasecmp(ncprg_on_pc,“o1000.mcv”) )
          { }
         else
          {
          printf(“⟨pre⟩The NC filename ⟨code⟩⟨font size =+6 COLOR=BLUE⟩ %s⟨/font⟩”,ncprg_on_pc);
          printf(“ is invalid⟨/code⟩⟨/pre⟩”);
          exit(1);
          }
  }
  else if (!strcasecmp(entries[x].name, “Operation”)) { strcpy(m_operation,entries[x].val);   }

  else {}
}
  if (!strncasecmp(m_operation, “cut”, 3) || !strncasecmp(m_activity, “status”, 6) )
```

```
    { /* The following output actuates client pull */
      /* where the file checkrmc.html is updated by the control program of the Machining center */
        printf("⟨META HTTP-EQUIV=REFRESH CONTENT=\"6;=http:/~rmc/checkrmc.html\"⟩\n");
        printf("⟨HR⟩This link uses client-pull to retrieve the up-to-date status of the RMC at CAMR");
    }
printf("⟨hr⟩");
strcpy(out_name,"/home/home2/camr1/yungchou/httpd/cgi-bin/RMC/rmc/rmc/mcv-1000.dat");
    fp = fopen( out_name, "w+");
    if (fp == NULL)
    {
        fclose(fp); printf("⟨pre⟩......cannot write into mcv-1000.dat......⟨//pre⟩");
    }
    else {
        fprintf(fp, "%s", m_name);
        fprintf(fp, "%s", m_loading);
        fprintf(fp, "%s", m_workpiece);
        fprintf(fp, "%s", m_bench);
        fprintf(fp, "%s", m_clamping);
        fprintf(fp, "%s", m_robot_function);
        fprintf(fp, "%s", m_activity);
        fprintf(fp, "%s", ncprg_on_pc);
        fprintf(fp, "%s", m_operation);
        fprintf(fp, "\n");
        fclose(fp);
    }

/* welcome to RMC */
    strcpy(String_to_RMC, "⟨APPLET CODEBASE=\"http:/yungchou/RMC\" code=\"playaudio.class\"
width=30 height=30⟩");
    printf("%s\n", String_to_RMC);
    strcpy(String_to_RMC, "⟨param name=model value=\"/~rmc/RMC_voice/");
    printf("%swelcometormc.au\"⟩\n", String_to_RMC);
    strcpy(String_to_RMC, "⟨/applet⟩");
    printf("%s\n", String_to_RMC);
/* ------------------------------------------------ */
    printf("⟨/ul⟩%c",10);
    printf("⟨/HTML⟩%c",10);
}
```

BIBLIOGRAPHY

1. J. M. Hawkins and R. Allen, *The Oxford Encyclopedic English Dictionary,* Oxford: Clarendon Press, 1991, p. 1222.
2. http://mgliris2.ecn.purdue/~toung/SONOMA/background.html
3. S. Puri, The impact of globalization on American industry, *Proc. 12th Int. Conf. CAD / CAM Robot. Factories Future,* Middlesex University, London, England, 14–16 August 1996, pp. 466–470.
4. S. R. Hiltz and M. Turoff, *The Network Nation—Human Communication Via Computer,* Reading, MA: Addison-Wesley, 1978.
5. L. Press, Before the altair: The history of personal computing, *Commun. ACM,* September, **36** (9): 27–33, 1993.
6. E. Krol and E. Hoffman, *FYI on "What is the Internet?"* ftp://nic.merit.edu/documents/fyi/fyi20.html, 1993.
7. A. S. Tanenbaum, *Computer Networks,* 3rd ed., Englewood Cliffs, NJ: Prentice-Hall International, Inc., 1996, pp. 47–54.
8. S.-H. Kang et al., Collaborative design using the World Wide Web, Technical Report TR97-02, IOWA Internet Laboratory, Department of Industrial Engineering, University of IOWA, USA. URL: http://128.255.23.188Internetlab/Techrep/tr9702.pdf, 1997.
9. N. Kim et al., 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. URL: http://128.255.23.188/Internetlab/Techrep/tr9704.pdf, 1997.
10. P. G. Ranky, Computer networks for world class CIM systems, *World Class Computer Integrated Manufacturing (CIM) Series,* Guildford: CIMware, 1990, pp. 37–38.
11. L.-R. Lin and H.-P. Huang, Real-time networking for the implementation of CIM, *Proc. 4th Int. Conf. Autom. Technol., Autom. '96,* **1**, Mechanical Industry Research Laboratories/ITRI, Hsinchu, Taiwan, R.O.C., July 8–11, 1996, pp. 21–28.
12. Y. C. Kao and Grier C. I. Lin, Accessing a remote machining cell through the World Wide Web, Accepted for publication by the *CIRP Int. Symp.—Advanced Des. Manuf. Global Manuf. Era—* August 21–22, 1997, Hong Kong: The University of Hong Kong.
13. G. R. Liang, Advent of network-integrated manufacturing age in Taiwan, *Proc. 4th Int. Conf. Autom. Technol., Autom. '96,* **1**, Mechanical Industry Research Laboratories/ITRI, Hsinchu, Taiwan, R.O.C., July 8–11, 1996, pp. 29–36.
14. J. Harrington, *Computer Integrated Manufacturing,* Huntington, NY: Robert E. Krieger Publishing, 1973.
15. M. P. Groover, *Automation, Production Systems, and Computer Integrated Manufacturing,* Englewood Cliffs, NJ: Prentice-Hall International, Part IV, 1987.

16. U. Rembold, B. O. Nnaji, and A. Storr, *Computer Integrated Manufacturing and Engineering,* Reading, MA: Addison-Wesley, Chapter 1, 1993.
17. C. M. Savage, *Fifth Generation Management,* Rev. ed., Boston: Butterworth-Heinemann, 1996.
18. http://www.gensym.com.
19. G. R. Liang, A comparison of four MI diagram representations in automated manufacturing system design, *Proc. CIIE National Conf.,* Tainan, Taiwan, ROC, Dec. 1993, pp. 554–561.
20. FANUC, DNC2 Descriptions, B-61992E/01, FANUC Ltd, 1991.
21. ABB Robotics Inc., Product Manual—IRB 2000, 3HAA 3907-101, January 1992/M92, Asea Brown Boveri, http://www.abb.com
22. ABB Robotics Inc., *CommTools User's Manual,* RobotWare, 1992, Asea Brown Boveri.
23. T.-D. Lu and G. C. I. Lin, CAD, vision, and sensor based intelligent robot server, *J. Comput. Integ. Manuf. Syst.,* **9** (2): 91–100, 1996.
24. http://hoohoo.ncsa.uiuc.edu
25. Y. C. Kao and G. C. I. Lin, CAD/CAM collaboration and remote machining, *J. Comput. Integr. Manuf. Syst.,* **9** (3): 149–160, 1996.
26. R. Juhasz, *XFS32 Version 1.0 Network File System (NFS) Client* Windows version Computer program. September 21, Copyright © 1994.
27. National Center for Supercomputing Applications, The common gateway interface, URL: http://hoohoo.ncsa.uiuc.edu/docs/cgi/overview.html
28. A. Ford, *Spinning the Web—How to Provide Information on the Internet.* New York: International Thomson Publishing, 1996, pp. 61–69.
29. http://www.w3.org/pub/WWW/
30. Netscape Communications Corporation, An exploration of dynamic documents, URL: http://www.netscape.com/assist/net_sites/pullpush.html, 1996
31. C. Musciano, How to get started with server-side includes, URL: http://www.re-dev.com/netscapeworld/nw-01-1997/nw-01-ssi.html, 1997.
32. M. Campione and K. Walrath, The Java tutorial, URL: http://www.javasoft.com//nav/read/Tutorial/index.html, 1997.
33. http://www.netscape.com
34. P. Wright, S. Schofield, and F.-C. Wang, Open architecture control for machine tools, http://infopad.eccs.berkley.edu/~fuchung/papers/Canada96/canada96.htm.
35. P. Raulefs, The virtual factory, *13th World Comput. Congr. 94,* **2**, 1994 IFIP, pp. 18–30.
36. P. Frohlich et al., Diagnostic agents for distributed systems, http://www.kbs.uni-hannover.de/paper/96/ma1.ps

G. C. I. Lin
Y. C. Kao
University of South Australia