

**A FEASIBILITY STUDY FOR THE IMPLEMENTATION OF
NON-SITE BASED HANDS-ON CURRICULUM FOR
ENGINEERING TECHNOLOGY EDUCATION**

Immanuel Edinbarough, Manian Ramkumar

The University of Texas at Brownsville / Rochester Institute of Technology

Abstract

This paper discusses the development of an intelligent architecture for web-based programming & control of an automated manufacturing cell. The cell consists of a three-axis TERCO CNC milling machine, an IBM SCARA robot and a raw material feed station. The cell components are integrated using a data acquisition system (DAS), programmed using Visual Basic and controlled through the web. A dedicated web page provides the users the opportunity to learn CNC programming and its control from remote sites. When the user submits a CNC program, the server in the lab takes control and executes various routines to complete the machining of the part automatically. The remote user can witness the operation of the cell through a real video feedback on the web page. Details pertaining to the development, integration and remote operation of the automated manufacturing cell is presented in this paper. The feasibility of providing non-site based hands-on engineering and technology education is also explored through this project.

I. Introduction

The advent of the world-wide-web has tremendously influenced the way in which industry and academia perform various activities. More and more academic institutions are starting to recognize the vital link that the web provides between the faculty and the students. As the web savvy students demand specialized attention for customized curriculum and training, it becomes the responsibility of the academia to provide new information technology based solutions, to satisfy these needs. In this regard, an attempt has been made to develop an automated cell capable of providing non-site based hands-on course in engineering and technology education. The main challenge in this effort is the remote programming and control of the robot and CNC machine. Details pertaining to the intelligent control architecture, system programming, information interchange, networking aspects of the cell and the issues related to the non-site based hands on training are presented in this paper.

II. Background:

The possibility of web-based education has opened endless opportunities in e-training and e-learning. There are numerous web-based training programs already available but the subject content is limited to computers and web development. Popular programs currently in use include: Kovacs Consulting, Internet & World Wide Web Trading (www.kovacs.com), dyroweb (www.dyroweb.com), MindLeaders (<http://onlinebasedtraining.com>). The use of the web as a training medium for lab-based courses still faces a myriad of challenges. Several educational institutions are currently working to address this problem and are still in the developmental stages. Web-based training programs which address the issue of equipment training such as CNC and robot programming, through the internet, are still considered new ideas and need consistent effort to make it a viable solution. The work presented in this paper will address some of the critical issues and some basic tools currently available for implementing this technology. New and emerging tools will provide incredible opportunities to expand this technology to new levels, in the near future.

III. Hardware Aspects of the Cell

The automated manufacturing cell consists of three fundamental automation units, which include an automated machining system, an automated material handling system and a data acquisition & control system. The hardware and design aspects of these systems are discussed in detail in this section.

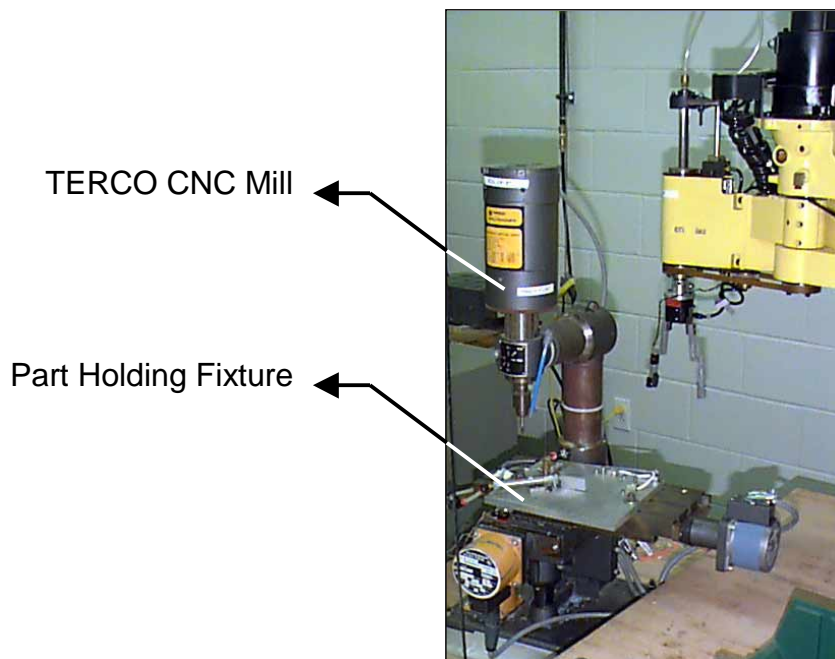
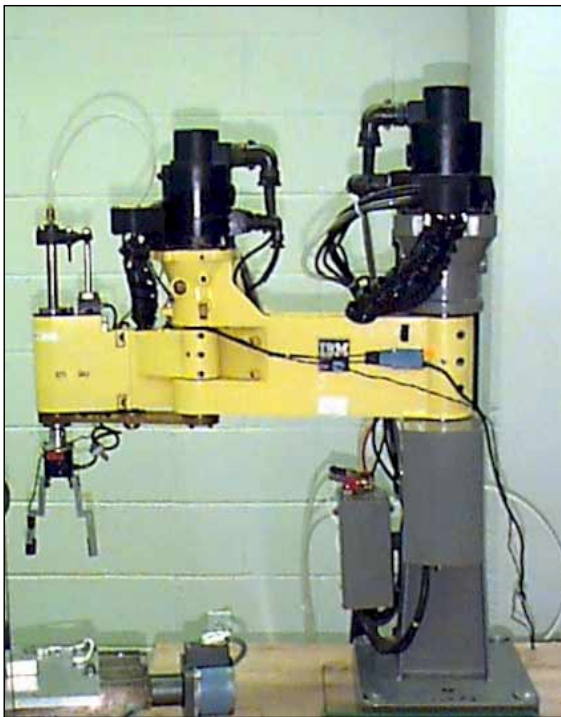


Figure 1. Automated Machining System

Automated Machining System

The automated machining system consists of a table top, three-axis TERCO CNC vertical milling machine, equipped with an automated part holding fixture (Figure 1). This system is used for engraving any pattern, developed by the remote user, to the specified size requirements. It operates at nine different speeds and can be programmed using standard G & M-codes. The machine has the capability for automated program download and digital I/O integration with external devices.

The automated part holding fixture is a pneumatically operated clamping device, capable of holding a block of raw material, 3"x2"x1.5" in size. The clamping mechanism is actuated from the mill controller. The fixture is fitted with fiber optic sensors, to detect the presence or absence of a part and if the clamp is closed or open.



IBM Robot for Load/Unload



Raw Material Feed Station

Figure 2. Automated Material Handling System

Automated Material Handling System

The automated material handling system consists of a robot and a raw material feed station (Figure 2). The robot used in this application is an IBM 7535 four-axis SCARA robot with non-servo Z-axis control. The robot has a stand-alone controller programmed using the AML language. The controller provides digital I/O interface for integrating the robot with other components in the cell. The robot is used for loading and unloading the part, to and from the

milling fixture. There is a micro-switch embedded in the gripper, to detect presence or absence of a part during handling.

The raw material feed station is used to feed the blocks of raw material, consistently, for robot pickup. It uses a simple gravity feed arrangement and has a micro-switch that detects the presence or absence of the raw material. The part feeder has the capacity to hold a maximum of six parts.

Data Acquisition & Control System

The data acquisition and control system is the brain behind the operation of the work-cell (Figure 3). It acquires information from the various sensors and also the digital I/O interface units of the CNC machine and the robot. The DAS used in the cell is an Optomux system manufactured by Opto 22. It consists of several input and output optical isolators (modules), which communicate with the Optomux brain board and links the information available in the field devices to the control program. The output modules are used to activate or deactivate field devices such as motors, solenoids, lights etc. and also provide signals to the CNC machine and robot. The input modules, on the other hand, keep track of the status of the cell through information provided by the sensors. The control program is written in Visual Basic, as it provides excellent graphical user interface for stand-alone operation of the cell.

The other data acquisition system used in the cell includes a video camera and a video capture card. This is used for providing real-video feedback of the operation of the cell, to the remote user. The feedback involves streaming of live video images on the web using real video producer, from the lab.

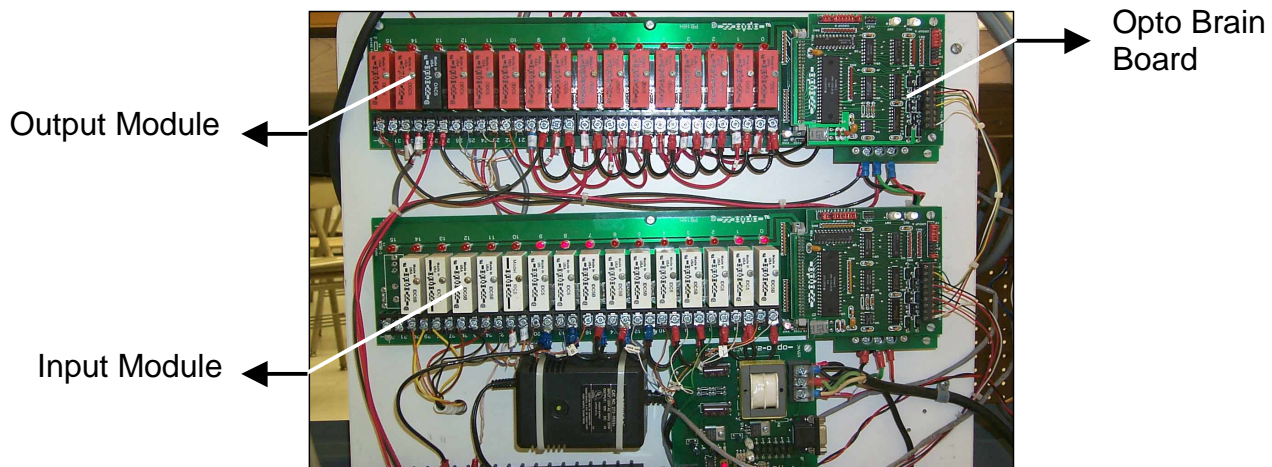


Figure 3. Optomux Data Acquisition & Control System

Cell Layout

The most challenging aspect of the cell layout is the positioning of the raw material pickup location and the machining fixture load/unload location, within the work envelope of the robot. This is primarily due to the constraints imposed by the robot work envelope, its configuration and the limitations of the slide movement in the CNC machine. To overcome these limitations, a

special milling fixture was designed and positioned on the CNC machine table, such that the robot could perform its task without interfering with the machining system. Another important requirement in the layout was to keep the fixture and the raw material feed location in the same plane for the robot to reach. This was due to the robot's non-servo Z-axis, which can reach only two positions, either fully retracted or extended.

IV. System Integration Aspects of the cell

The various hardware components of the cell, discussed in the previous section, are integrated by the use of sensors and digital I/O signals. These signals are monitored by the DAS, which takes appropriate action based on the user program. The action is in the form of output module activation or deactivation. This provides control to the field output devices and signals to the robot and CNC machine. The user program, developed using Visual Basic, controls the various activities taking place in the cell and manages the overall integration of the components. This program makes use of custom routines and control objects provided by Opto 22. The flow of information between the hardware components of the cell and their integration is illustrated in Figure 4.

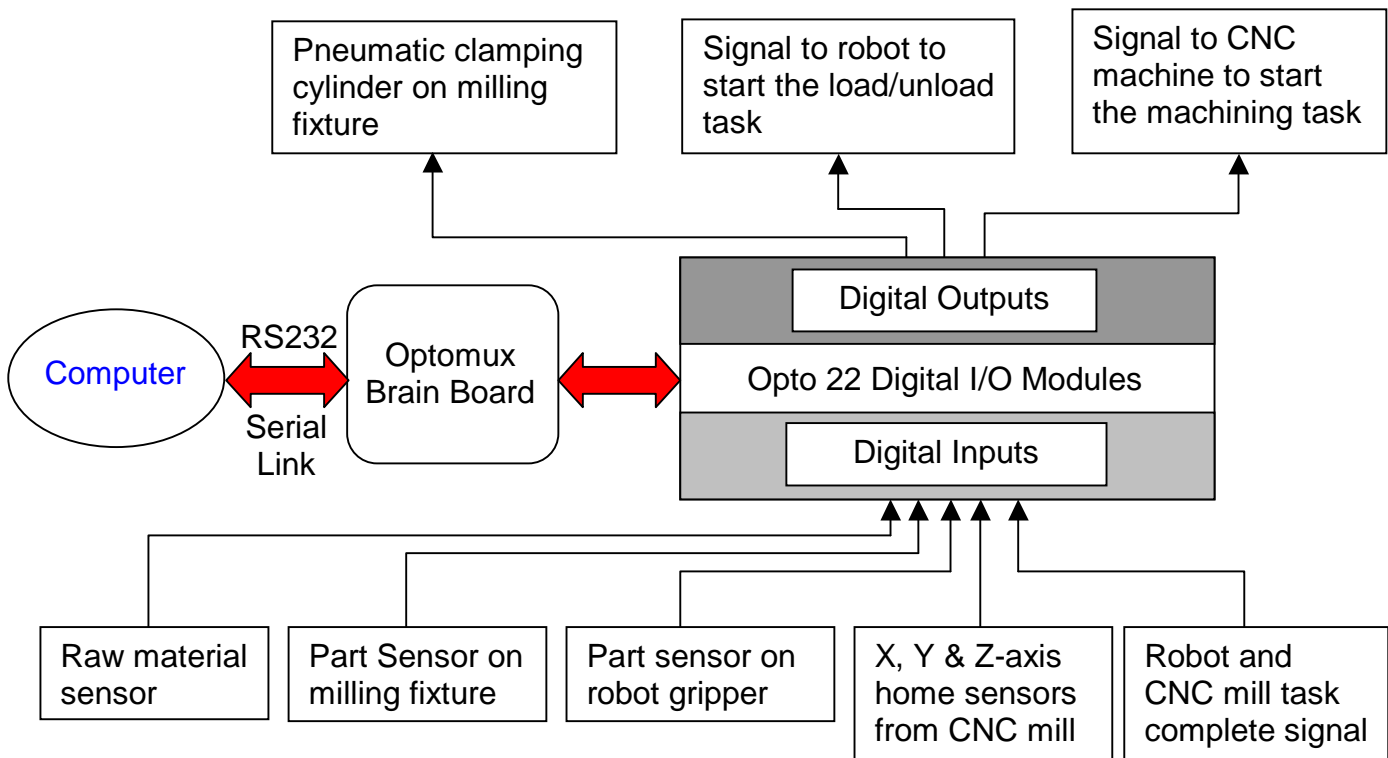


Figure 4. Information Flow Between Cell Components & System Integration

The system integration enables the cell control computer to perform the following sequence of tasks, automatically, when the user downloads a CNC program from a remote site.

- ◆ Check for robot and part availability

- ◆ Load user program into the CNC controller
- ◆ Instruct the robot to pick up raw material from feeder and place it on the machining fixture
- ◆ Instruct the CNC machine to engrave the pattern
- ◆ Instruct the robot to pick up finished part and place it in the finished part bin
- ◆ Provide feedback to the user

Details pertaining to sensors, digital input/output signals and intelligent control architecture are discussed in this section.

Sensors

The cell uses several contact and non-contact proximity sensors to collect the status information of the different systems. Apart from providing feedback information they enable fool proofing of the cell activities. The signals from these sensors are fed to the input/output modules of the DAS, for further processing. A listing of the various sensors used, their location and purpose are summarized in Table 1.

Table 1. Sensor Types, Their Location and Purpose

Sensor Type	Location		Purpose
Micro-switches (Contact)	A	Raw material feed station	Presence or absence of raw material
	B	Robot end-effector	Presence or absence of part in the grippers while handling
	C	Part holding fixture	To sense the alignment and presence of the part in the machining fixture
	D	CNC Milling Table XYZ Home	To sense the home position and table over travel of the machine
Induction Sensor (Non-Contact)	E	Part holding clamp	To sense the extended or retracted position of the clamping cylinder

Digital Inputs & Outputs

The CNC and robot controllers use digital I/O signals to coordinate their activities with the rest of the cell. These signals are connected to the input/output modules of the data acquisition and control system. They are generally TTL signals (5 V DC) or 12-24 V DC signals. The CNC machine and the robot controller monitor input signals and initiate output signals using specific commands, within each of their programs. Figure 5 shows a typical digital I/O signal exchange, between the robot and the DAS.

Intelligent Control Architecture

The cell control architecture consists of three hierarchical levels of data acquisition and control (Figure 6). The hardware components that constitute each level of the hierarchy are given below.

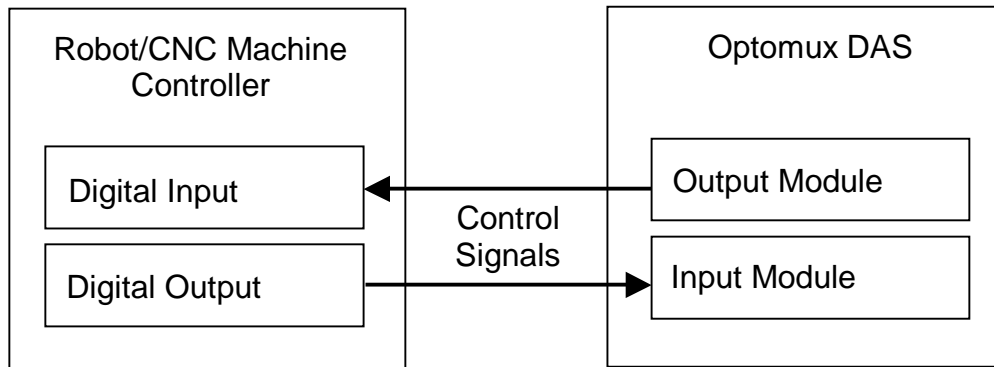


Figure 5. Digital Input/Output Communication between Robot/CNC Machine and DAS

Level 1. Sensors, Actuators and digital I/O interface

Video camera for feedback

Level 2. Data acquisition and control system to communicate with the sensors and digital I/O interface

Video capture card to acquire the video images from the camera

Level 3. A cell control computer, which communicates with the DAS, through the serial interface (RS232C)

Real-video server for interfacing with the video-capture card

These hardware components are stand-alone systems without any application specific intelligence. This intelligence is provided, to the systems, by writing application specific programs. The programs used in the cell control architecture include:

1. CNC and robot programs
2. Intelligent routines for controlling the DAS and operation of the cell (Visual Basic)
3. Web page that provides user interface, feedback and the capability for program development and download (HTML & CGI-Scripts)
4. Real video producer for video compression and streaming

Details pertaining to these programs are discussed in the next section.

V. Programming and control Aspects of the Cell

The cell control is primarily accomplished by the use of intelligent routines written in Visual Basic, CGI-Script and HTML. These routines along with the CNC machine and robot programs enable information exchange between the various levels of the control architecture. This forms the software framework for the cell (Figure 7). This section highlights the logic used within the individual control routines and their information exchange strategy.

Robot and CNC Control Programs

The stand-alone robot program developed using the AML language, controls the raw material loading and unloading operation. Since this sequence of operation is fixed, the user need not develop a new program. The program consists of motion commands and interfacing commands. The motion commands enable the robot to move to different positions and the interfacing commands enable handshaking of the robot controller and the DAS. This provides status information to the visual basic program and to the user. When the robot program is executed, it moves the robot to the home position and waits for a signal from the DAS, to continue execution of the programmed routine. Similarly, at the end of each run, the robot controller sends a signal to the DAS, which enables the Visual Basic to continue execution of the subsequent sequence of operation.

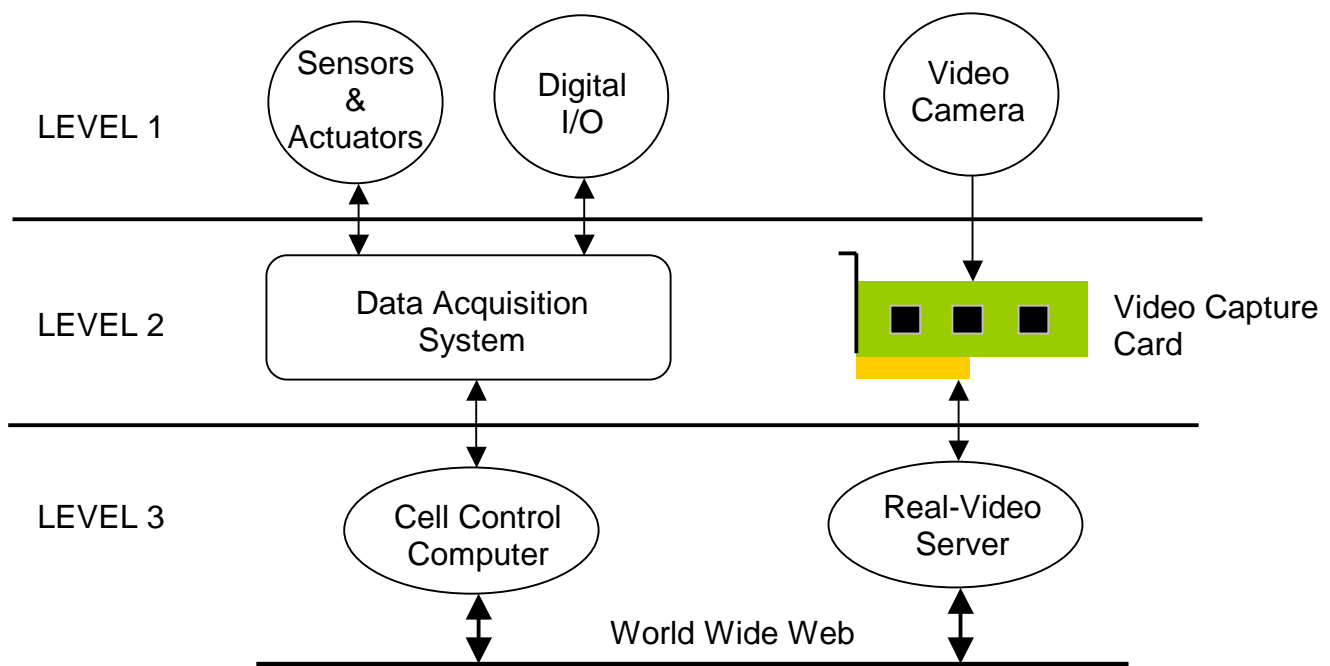


Figure 6. Intelligent Control Architecture

The stand-alone CNC program also consists of motion commands and interface commands. The motion commands enable the movement of the machine slides to perform the engraving operation. The interface commands, on the other hand, provide handshaking capability with the DAS. The user develops and downloads the CNC program using the editor provided in the web page. The web page also provides a tutorial for the user to learn CNC programming.

Visual Basic Program for Optomux DAS Control

The optomux DAS is monitored and controlled using a Visual Basic program. This program enables the main cell control computer to communicate with the DAS through the RS232C serial interface. The brain of the optomux DAS receives command strings from the program. These command strings contain information pertaining to the address of the system, input/output

module positions to be monitored and controlled and the specific control command to be executed. Examples of different control commands include activate, deactivate, read status, generate N pulses, read counter, read latch, etc. For each command issued by the cell controller, the DAS acknowledges receipt of the command and also its execution. If the command string is not generated properly, the system returns a not-acknowledged signal, which requires the user to take corrective action. The command strings can either be built by the user or by using function calls provided in the Optoware driver program, developed by Opto 22.

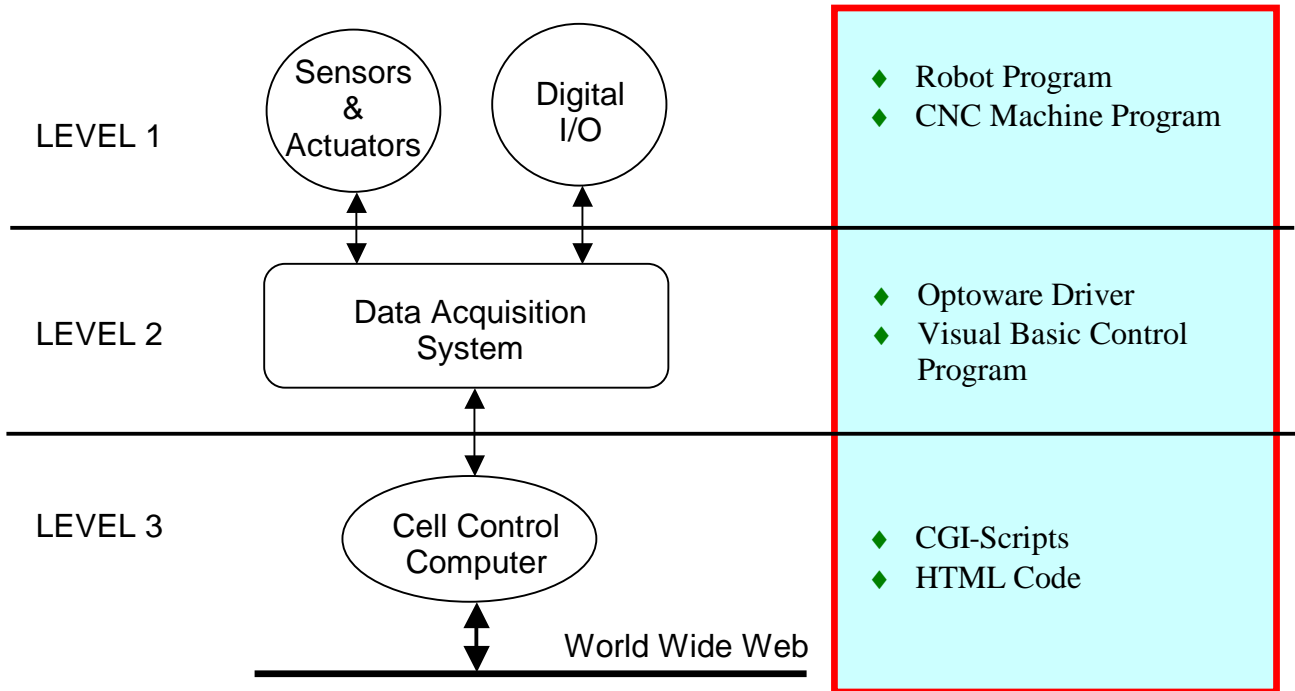


Figure 7. Software Framework for the Cell

The Visual Basic control program is the most crucial component of the software framework. This is due to its importance in performing various control and information exchange activities required in the cell. These activities are as follows:

1. Receiving the CNC program and the cell operation instructions from the user input, through CGI-Scripts and the web page
2. Downloading the CNC program
3. Controlling the DAS
4. Coordinating the various subsystems and their activities in the cell
5. Reporting status information of the cell back to the web page

Sample Visual Basic programs to perform CNC program download and DAS control is provided in Figure 8.

PERL Script for Data Exchange Between Web and Cell Server

Perl is a high-level interpreted programming language used for writing Common Gateway Interface (CGI) scripts. These scripts are used to collect user inputs from the web page and forward them to the visual basic control program and also to maintain fail proof operation of the information interchange (Figure 9). A web server, also the cell control computer, hosts the web pages that the remote user will access. A program editor available in the web page allows the user to develop and download CNC programs. When the user activates the download button on the page, the action is transferred to the main CGI-script, which authenticates the user access and initiates the visual basic control programs. The flowchart for the various tasks executed by this script is provided in Figure 10.

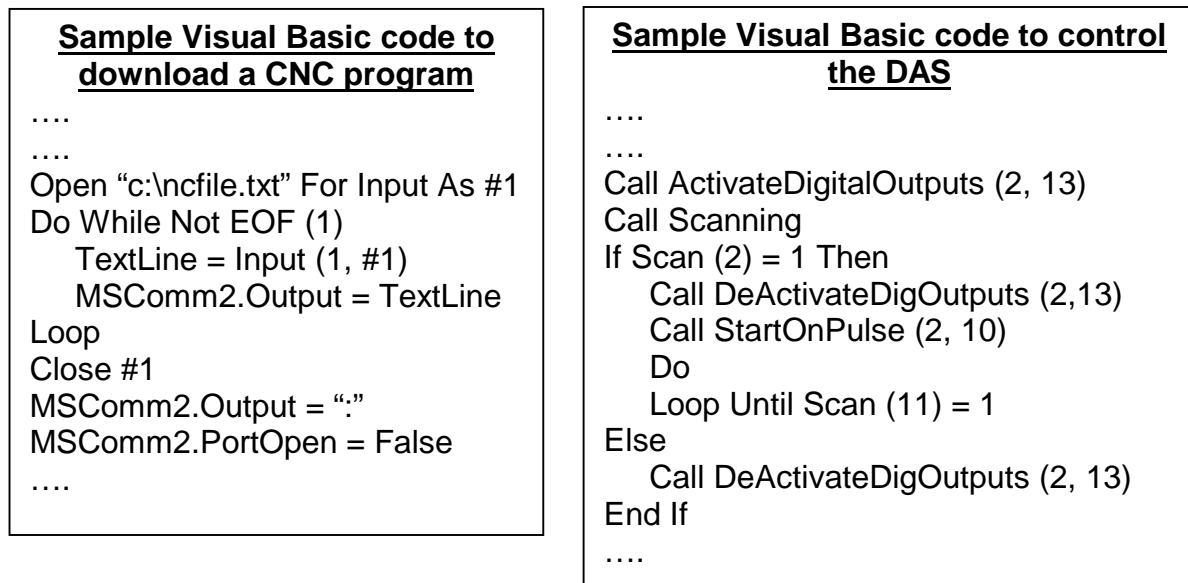


Figure 8. Sample Visual Basic Programs

HTML Scripting for User Program Download and Control

The main control computer for the cell also acts as the web server. It contains the html documents and the CGI scripts. The web address for the server is <http://mnet-cimlab18.cims.rit.edu>. The web page design provides the following features:

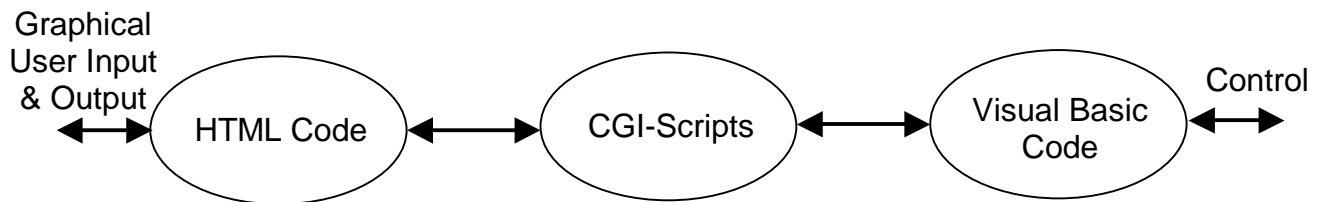


Figure 9. Information Interchange between Programs

1. Cell Description and Layout
 2. CNC Programming Tutorial
 3. Automated program development by choosing alphabets (programs for each alphabet already available)
 4. Program development by the user for their pattern
 5. Program download and control
 6. Program execution and feedback.
- A sample HTML source code is shown in Figure 11.

VI. Benefits and Issues in Non-site based Hands-on Courses:

The following benefits will play a major role in sustaining the student's interest and their retention in non-site based hands-on courses:

1. Greater communication between faculty and students
2. Partnership between instructors on site and instructors at other institutions
3. Latest information is communicated at faster rate
4. Course delivery becomes more easy and less expensive
5. Duplication of effort is eliminated
6. Learning materials and other resources can be cross-referenced with hyperlinks
7. Technical support requirement is reduced
8. Inclusion of graphics and animation will give depth to the content and material

Even though the above list of benefits is impressive, web-based delivery of hands-on courses requires a thorough understanding of the following issues:

1. Student motivation
2. Training and adaptation time required with new technology
3. Problems with new technology implementation (should be addressed effectively and on time)
4. Limited instructional methods because of current hardware and software limitations
5. Equipment operational safety
6. Security issues

VII. Conclusion & Future activities

The above project was undertaken to develop the infrastructure to attain the goal of establishing a fully operational non-site based lab course. The success of this effort has provided us with enough motivation to proceed further. This project has helped us build the following infrastructure:

- ◆ Development of a remotely controlled manufacturing cell that can be utilized for web based hands-on CNC training
- ◆ The integration of sensors and digital I/O interfaces with a computer based data acquisition & control system in developing an intelligent control architecture
- ◆ Development of intelligent routines using visual basic, CGI-scripts and HTML for web based cell programming and control
- ◆ A framework for future hardware and software enhancements that will lead to the development of a more advanced web based automation training system

Some of the future activities to be undertaken include:

1. Improved fool proofing of the cell
2. Providing enhanced feedback to the user regarding cell status
3. Incorporating advanced CNC capability and programming
4. Implementing robot program development and download feature on the web
5. Including vision based quality check and feedback

Bibliography

1. Sensor Product Selection Guide, Honeywell Inc.
2. Numatics Product Catalog for pneumatic cylinders and actuators
3. IBM Robot and TERCO CNC Milling machine programming and maintenance manuals
4. Optoware Driver Reference
5. "An Introduction to Programming using Visual Basic 5.0", 3rd Edition, David I. Schneider, Prentice Hall, 1998
6. "Mastering HTML 4.0", D. S. Ray and Eric J. Ray, BPB Publications, 1998
7. <http://cgi.resourceindex.com/>

Immanuel Edinbarough

Immanuel Edinbarough is a Professor in the department of Engineering Technology at the University of Texas at Brownsville. He has 6 years of industrial experience in the field of machine tools manufacturing. He has been in the teaching profession for the past 13 years, 3 years as a visiting professor at the Rochester Institute of Technology. His teaching and research interests are in the areas of Automation, Robotics, Machine Vision, and CAD/CAM/CIM. He has published several papers, in these areas, in various national & international conferences and journals.

Manian Ramkumar

Manian Ramkumar is a faculty in the Computer Integrated Manufacturing Engineering Technology Department at the Rochester Institute of Technology, Rochester, NY. He teaches courses in CAD, CAM, CIM, FMS, robotics, surface mount electronics manufacturing, assembly automation, and controls for manufacturing automation. He was instrumental in developing the CIM and Surface Mount Electronics Manufacturing laboratory at RIT. These laboratories are equipped with production scale equipment that is used for hands-on training and conducting applied research projects for companies.

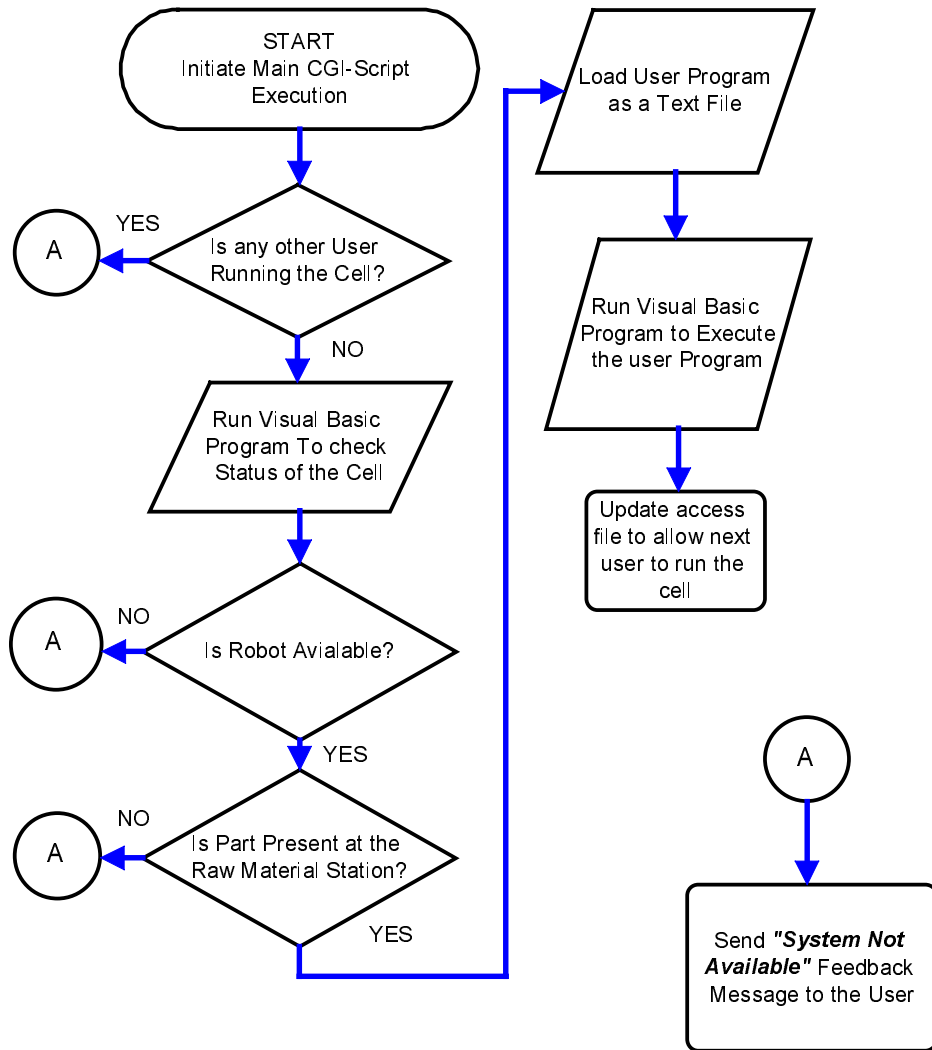


Figure 10. Flowchart for Tasks Executed by the CGI-Script

```
.....
.....
<p>Enter your program and click the upload button to transfer the file to the CNC controller. But.....</p>
<p align="left">read the
<a href="../docs/tutorial.htm#how">tutorials</a> for more information on
uploading files. Please follow the instructs on how to download a program. </p>

<p align="left">Please enter the program in the text area below or click below
to go to make a selection from previously written programs.</p>

<p align="center"><a href="../docs/selection.htm"></a></p>
<p>&nbsp;</p>

<form method="get" action="../cgi-bin/nph-main.pl" name="Upload">
<p>&nbsp;</p>
<p align="center"><font color="#0000FF">NC-CODE:</font></p>
<p align="center">
<textarea name="textfield" cols="50" rows="5"></textarea>
</p>
<p>&nbsp;</p>
.....
```

*Main CGI-Script
call that performs
information
exchange and cell
control*

Figure 11. Sample HTML Source Code