

# Real Laboratories for Distance Education

---

Stuart McCracken, Zeljko Zilic and Hoi Yun Henry Chan

McGill University, Montréal, Canada

Providing distance laboratory-based courses is becoming critical for distance technical education. In this work, we describe remote laboratories in digital system courses. While the hardware is based on widely used programmable logic, the Internet interfaces include those for remote development, testing and debugging as well as the cooperative work environment. Special attention has been paid to the objectivity of evaluating the remote cooperative work. The web tools for project progress evaluation, self- and group-assessment and the automated hardware support are being developed. Previous work consisted mainly of providing simulated environments or prefabricated circuits. The productivity and accessibility of these tools was greatly enhanced by using off-the-shelf hardware, software and networking elements.

*Keywords:* distance education, remote laboratories, field-programmable gate arrays, peer evaluation, user interface, video collaboration, videoconference.

## 1. Introduction

Distance education has experienced great improvements with the advents of Internet. Much of the “low-hanging fruits” have been consumed by now: distributing and managing lectures, examinations and grades has become transparent and equivalent to that of on-campus education. The areas that still need improvement include those with a substantial technical content that can only be provided with running comprehensive laboratories. Incidentally, the greatest increase in educational needs have been in the engineering areas that address large collaborative system developments [12].

Our goal is to investigate the practicality of having remote laboratories for digital design engineering courses. These remote laboratories allow any user in a remote location to program or

test a system under development. Furthermore, with software intervention, multiple users could be using, viewing and discussing the same program and a digital circuit under development. The actual outputs of the electronic circuit could be made visible to all the developers, as well as to a teacher. As teleconferencing tools have become readily accessible, we provide means to use commercial off-the-shelf (COTS) components [14][15][16].

From a hardware point of view, Field Programmable Gate-Arrays (FPGAs) [10][13] can be effectively deployed in hardware design distance laboratories. This technology allows rapid development of digital circuits, by downloading a digital circuit onto a device. In a learning environment, the value of FPGAs has been shown to be indispensable. They can quickly be debugged, adapted and reprogrammed without physically manipulating the FPGA. Thus, FPGAs are very appropriate when considering remote laboratories in cooperative network environments. They further remove much overhead in running laboratories as one piece of equipment can serve in many experiments and projects. FPGA development systems can be installed in laboratory only once, while students and laboratory technicians can access them remotely at all times.

The FPGA use and their remote development system are considered here in the context of distance education for digital system design courses, including those with a large project component. Several approaches to establishing the needed infrastructure are evaluated with respect to their practicality, followed by description of our low cost remote laboratory that is easily accessible by any remote student with a

personal computer. Since much of the underlying teleconferencing technologies are still being under development, we chose to rely on widely available de-facto standards that allow the use of low bandwidth channels.

## 2. Background

### A. Related Work

Recently, several attempts have been made to establish University laboratories for use in distance education. Systems such as the AIM-Lab project [1], and the RemLab [2] have made headway in the measurement field. Equally fruitful advances have been made in the areas of robotics and control with such accomplishments as the visualization, simulation and control of a robotic system [3]. Furthermore, remote control laboratory experiments have also been developed in an Internet-based setting [4]. This paper attempts to accomplish similar results in the courses that teach design and implementation of digital systems, with emphasis on completing large-scale laboratory projects. This implementation will require the use of COTS software components integrated into one remote laboratory with multipoint access.

### B. Groupware and Performance Feedback

Historically, the interaction between users and remote systems falls into a category of 'groupware' research. The goal of this research, which dates back to the late 1980's, is to understand human communications in order to facilitate the collaboration and interaction of all users [5]. The study in this field is done on both a computer science and human perception level in order to understand how we, as a population, interact with each other and with computers.

The system considered here has many attributes of the Computer-Supported Cooperative Work (CSCW). CSCW strives to understand how technology can improve group work, whether it is in the same room, or halfway around the world [5]. In recent years, globalization trends forced many enterprises to engage in some form of CSCW. We are seeing increasing needs for new graduates that can work in teams spread in several locations. This fact alone justifies investigation of CSCW applications in building large

systems. Nonetheless, computer-supported collaboration still has much room to grow [6]. Among the impediments to wider CSCW acceptance, we notice practical interoperability and lack of integrative frameworks [18].

The research on groupware provides one component of this work; our other focus is on the genuine education needs in using distance laboratories. In our scenario, while cooperative development is necessary, much emphasis is placed on improving and evaluating student learning by means of increased feedback gathering.

### C. Overview of the Multi-User Distance Laboratory

The Multi-User Distance Laboratory (MUDL) was developed and implemented using COTS components. We developed, in short time and with minimal cost, a concrete and feasible system for distance education using a digital system laboratory. MUDL could allow students to take a digital hardware class, in which they would design, discuss, implement and debug their designs on remotely placed hardware. The key ingredients in this laboratory are programmable logic devices, such as FPGAs.

The devised system involves one station where the hardware is placed, and multiple user locations from where students access the system. Figure 1 contains a high-level overview of the system, where one station is serviced by X user locations. As shown, multiple users access one remote FPGA site and collaborate by working on the same FPGA attached to the terminal. Thus, to supply resources for several groups using a number of FPGAs, identical systems can be used for prolonged periods of time. Such flexibility in use is also valuable for managing and maintaining the laboratory equipment, especially since lab technicians can remotely monitor and diagnose all laboratory equipment.

## 3. Software Components

### A. Multiple-User Collaborative Environments

In the design of our system, the main challenge involves the implementation of an useful multi-user development environment composed of standard and widely available components. Currently, software such as MS NetMeeting™

[8] allows multiple users to exchange live video in a conference format. All users can view video streams from all other participants. Provided that they have video cameras, their faces and gestures can be used to enrich other means of communication. This software allows various other means of multi-used interaction, from a simple chat to more structured collaboration. Features of NetMeeting are typical for most existing teleconferencing systems.

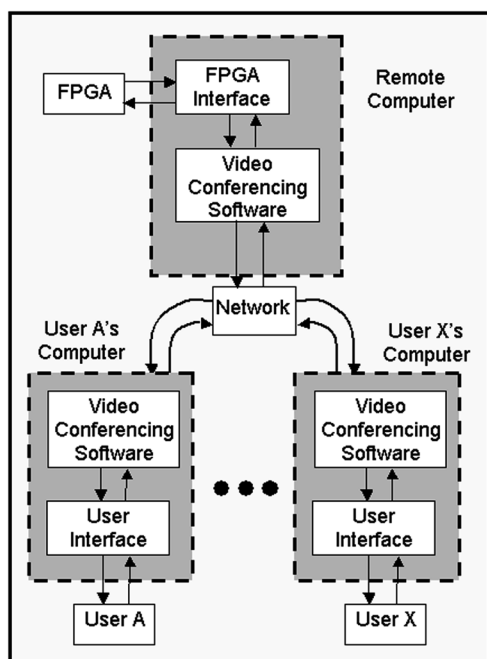


Fig. 1. Multi-User Distance Laboratory.

Equally applicable, AT&T research laboratory in Cambridge developed Virtual Network Computing (VNC) [9] software that allows remote users to view, access and work on a remote computer using standard protocols based on TCP/IP. Also, Microsoft developed their own software allowing for remote system use, called Remote Desktop Connection [17]. For the purposes of this system, only VNC's integration is considered. VNC's major benefit lies in its cross-platform nature. It is possible to view and execute all the commands of, for example, a Sun Workstation on an NT computer and vice-versa. This software displays the exact desktop of the accessed computer allowing for direct access to the remote computer. There are some current limitations to this software that shall be discussed later. Nonetheless, VNC provides a viable and impressive alternative to video.

One concern in the design of the MUDL is the principle of shared file access. For instance, two users cannot be working on the same file at the same time unless measures such as ownership and revision control can be incorporated. Nonetheless, supporting software can handle much of this burden. As much has been done on the subject, one of the pre-existing systems will suffice for the needs of this system. File sharing issues will be addressed in Section 6.

### B. FPGAs and FPGA Software

Most of the available FPGAs and relevant FPGA software are applicable to this system as they are based on the same principle. They require a software component for editing, compilation, simulation, and programming of the FPGA, and a hardware component that connects the terminal to the FPGA. Furthermore, the FPGA is mounted on a board that allows a simple off-the-shelf serial or parallel connection between terminal and FPGA.

For the uses of this design, the Altera University board (UP1) is used. This board contains two FPGAs both of which are programmable via the Altera Software Package [10] that we found robust and easy to use.

### C. Interfacing between the Internet and the Altera FPGA Software

A major missing link comes by way of the interfacing between the FPGA software and the CSCW software. The problem arises from the fact that the FPGA software, for the most part, was not designed with networking in mind. A software layer is used to fill the divide between the two. Furthermore, more functionality is allocated to software that will be referred to as "Interface Software". The interface software will also incorporate the display of Virtual LEDs and the use of Virtual DIP switches, which are common inputs and outputs on FPGA development boards such as the UP1, mentioned in the previous section.

Microsoft has made readily available a developer studio package for creating the interfaces between newly created software and the MS NetMeeting software. In order to accomplish such an interfacing, the newly created software

must have pointers that link to NetMeeting objects allowing access to the data channels between users that NetMeeting creates. This simplifies much of the network programming considerations.

Equally plausible as a solution is the use of a remote launching of software. If user A launches the FPGA software and works 'virtually' on the remote computer wired to the FPGA, the functioning of the Interface Software would be shifted from sending FPGA programming information to launching and controlling remote software. Furthermore, a combination of the two is also applicable.

VNC, on the other hand, does not require any such FPGA Software-to-Conferencing Software Interfacing Software as the FPGA software can be run directly from the remote computer.

If working on a remote system is slower than working on a local version of the software, it is more effective for the user to work on the software locally, and send the compiled files to be downloaded. Once the files are at the FPGA site, they are used to configure the FPGA and the circuit outputs are tested. This ensures that the resources are used more efficiently.

#### *D. Interfacing between Collaboration Tools and the FPGA Software Components*

As none of the mentioned collaboration software interfaces directly with the FPGA or the FPGA software, it is critical that such a link be made. We provide such interfacing software (IS). Equally, communication between the collaboration software and the FPGA is also done using IS. Functionality of IS necessarily relies on the solution to the previous problem, Sec. 3.C. If the remote computer receives programming information, IS needs to receive the data and pass it on to the right programming device (COM port, serial port or other). If, on the other hand, the remote computer receives commands to launch programs, it must be set up to launch programs and be able to use incoming information to properly run programs and use its resources.

Within NetMeeting, program sharing is possible which allows software to be used by larger student groups (4 students, as required by the course) including the remote computer. This,

however, is not as practical as VNC, which allows a user to view and work directly on the desktop of the remote computer.

#### *E. FPGA Input and Output Pin Control and Probing*

Once an FPGA is programmed, testing the outputs and simulating inputs is done with Interfacing Software. Without this functionality this entire FPGA remote system is not fully functional. Hence, it is critical that this portion be effective and takes into consideration as much of the FPGA's uses as possible in order to satisfy all users. Again, the solution to this problem lies in the two previous implementations. Ultimately, the interface must contain all standard FPGA development board functionality such as DIP switches, LEDs and pushbuttons. Equally important is the display of certain programming signals such as the 'Done' signal, which indicates when the file has been completely downloaded onto the FPGA. Many solutions are possible for the selection or display of the signals. Certain options will be discussed in the implementation section.

## 4. Software Assessment

We evaluated NetMeeting and VNC and discussed possible scenarios of their use. Their usability and system relevancy in terms of their benefits and drawbacks are given next.

### *A. NetMeeting*

#### *1) Benefits*

NetMeeting's major benefit is its ability to allow multiple user video conferencing. With this functionality, multiple users can view a remote FPGA system with a camera. All these users can also view each other in a conference format. Furthermore, NetMeeting is flexible as it allows for third party use of its data channels, which can improve functionality and applicability. Furthermore, NetMeeting has incorporated ftp and chat functionality. Thus, its main advantage is its ability to provide quality real-time video, and an all-encompassing environment for discussion and file sharing.

## 2) Drawbacks

The major drawback of NetMeeting is the increased level of complexity inherent to the introduction of third party software. Although this added complexity could be beneficial, other systems accomplish similar ends with alternative means.

### B. Virtual Network Computing (VNC)

#### 1) Benefits

VNC works as a cross platform remote viewer. It allows for a Unix workstation to be viewed by an NT desktop and vice versa. Equally important is the ability with which one can control the remote computer's resources as the user is working directly on the remote desktop. The main concept behind this software is that a user can fully control the remote environment, which inherently allows for the use of all the remote hardware that is connected to this remote terminal. Essentially, the desktop can be accessed with the VNC Viewer, or with any web browser. This allows VNC to fit well within the COTS component criterion stated previously.

#### 2) Drawbacks

VNC has no provisions for videoconferencing. If a camera display of the FPGA is shown on the desktop, it is relayed to all VNC viewers. All the viewers, however, are not able to conference with each other with VNC alone. Furthermore, as VNC is a cross platform system, 'drag and drop' file transfers are not possible due to operating system's differing file naming format and file structures. Finally, as Windows is not a multi-user system only one user can view the desktop through VNC.

#### C. Summary

As both NetMeeting and VNC have benefits and disadvantages, an approach using a combination of the two programs allows for the best results concerning multi-user remote FPGA programming. VNC's ability to easily work on a remote desktop provides a powerful alternative to the complex programming required with NetMeeting. NetMeeting does, however, incorporate live video very effectively, which also proves vital to the conference nature of this system.

## 5. Implementation

In this section, three possibilities as to the use of the NetMeeting and VNC are investigated. The first option relies on VNC alone, the second, on NetMeeting alone, and the third option combines the benefits of both software packages.

### A. Option 1

Option 1 consists of simply using VNC as the means to achieving the goal. Using VNC and a camera on a Unix machine allows for all users to see the procedures taking place on the remote computer. Furthermore, the camera allows for the FPGA board to be seen by all users. There is, however, no conferencing with the other participants. One alternative is to supply a notepad on the remote computer so that any user can jot down their thoughts. This is, nonetheless, very rudimentary and is preferably avoided. Equally, additional software is placed on the remote desktop that allows interfacing with the FPGA via DIP switches and LEDs. The layout on the user's desktop looks like that seen in Figure 2a.

### B. Option 2

The second option consists of using NetMeeting only. With this option, sophisticated user end and remote end interface software are required to perform the task effectively. This option, however, is an improvement over option 1 as video conferencing is now possible. All other users can chat and see each other with NetMeeting. Also, software at the remote end does not need to be launched as it can all be launched from the user's computer with NetMeeting. If only video cameras are available, a chat window will supply a means for communication between all the users. If, on the other hand, the camera (and microphone) handles audio and video, the chat window is not needed. The layout for such a set-up can be seen in Figure 2b.

### C. Option 3

Option 3 relies on a combination of both VNC and NetMeeting. VNC handles the FPGA interfacing and NetMeeting handles the conferencing aspects of the system. Ultimately, the layout for this option is shown in Figure 2c. A more in-depth description of this system follows in the next section.

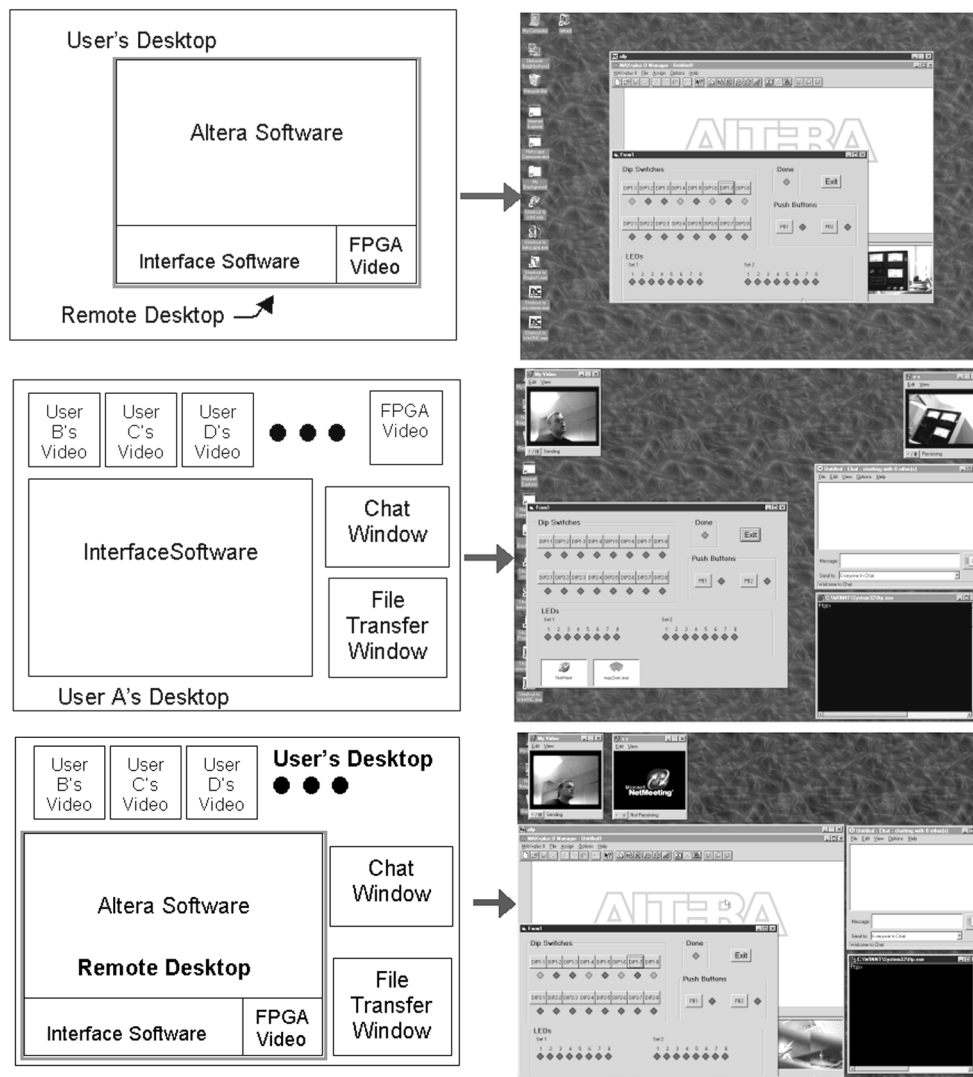


Fig. 2. Screen Shots and Descriptions of the 3 Optional User Environments;  
a) VNC Only; b) NetMeeting Only; c) NetMeeting and VNC.

## 6. Implementation of the Multi-User Distance Laboratory Using VNC and NetMeeting

The most effective system will be the one that combines both VNC and NetMeeting as seen in Figure 2c (Option 3). The addition of a third party interfacing software makes for a fully comprehensive system allowing for testing and implementing designs on the FPGA board. This is the system that was implemented and proved useful.

### A. Relevant NetMeeting Features for Our Implementation

NetMeeting's main use lies in its ability to conference live video. Furthermore, the chat and

file transfer functions are also useful, as this is required of the FPGA software system. They are very straightforward to use as they work like standard Windows™ programs. Once the program is opened, a connection can be made to the remote computer. As the setting on the remote computer is to accept any incoming calls, the connection would be made automatically. This setting is simply set in the Options → General window. Once the connection is made, the video channel must be opened. This is done in order to send to and receive from the other participants and the FPGA camera the live video.

Furthermore, the chat window can be opened straight from the pull-down menus (or ctrl-T).

Equally straightforward is the file transfer function. The sender would access the file transfer function via the pull-down menus (or cntrl-S) and select the file to be sent. The remote FPGA computer receives the file where it is saved in a specified directory. The user can then access, use and move the file with VNC. If the user wants to retrieve a file, this can equally be done with VNC by ‘ftp’ing the file back to his computer. VNC does not directly have an ftp function, but as the entire desktop and hard drives are available to the user, an ftp program can be invoked and used. Equally possible is to write a program that, using NetMeeting’s data channels, allows for direct drag-and-drop access to the remote hard-disk and other file storage utilities.

*B. Relevant VNC Features for our Implementation*

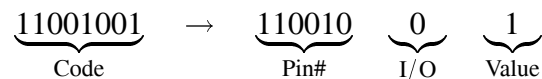
VNC is useful when it comes to the direct implementation with the FPGA. The remote computer requires the installation of the VNC program and each user-end computer requires the installation of the VNC viewer. The program is simple to use. When starting the program, the computer address to be connected to must be entered along with the password. The remote desktop is then loaded and used as if the user were there.

Within the confines of the VNC window, Altera software, the FPGA video and a third party software have been added. Altera software is used as if the user were working on his own desktop. This requires that the FPGA be always connected to the computer. To program the FPGA, the regular download program is used. The FPGA video also needs to be on at all times so that the user can view it. Finally, the third party software allows for using and testing of the circuit that was downloaded onto the FPGA.

*C. Interface Software*

In order to interface between the computer and the FPGA, a program was devised. This program allows for much of the functionality available on the UP1 Altera board.

Different solutions allowing for the passing of data between the FPGA’s pins and the computer terminal are possible. One example is to have a select number of pins that are probed constantly for their output. Some others are dedicated input pins. These are accessed by means of serial and parallel ports. This, however, only allows for a certain number of pins to be probed and used at once. If this is sufficient, this is the ideal alternative. If, on the other hand, many input and/or output pins are required, some additional circuitry is added to the FPGA in order to shift in the data to the pins. If the downloaded circuit does not take up all of the FPGA’s logic cells, this extra circuitry could be added to the design. For instance, each input is sent to the FPGA with one extra code for its pin number, one extra code for whether it is an input or output value, and the last code could dictate the actual value assigned. For instance:



A sample of the output of this circuit can be seen in Figure 3. The circuit shifts in the inputs, decodes the values, and assigns the correct input value to the correct pin. During this time the disable signal goes high, which disables the circuit being tested as these transitions could give erroneous results. This does, however, require design changes for a remote version as compared to the hands-on version. The remote version requires an external disable pin. This

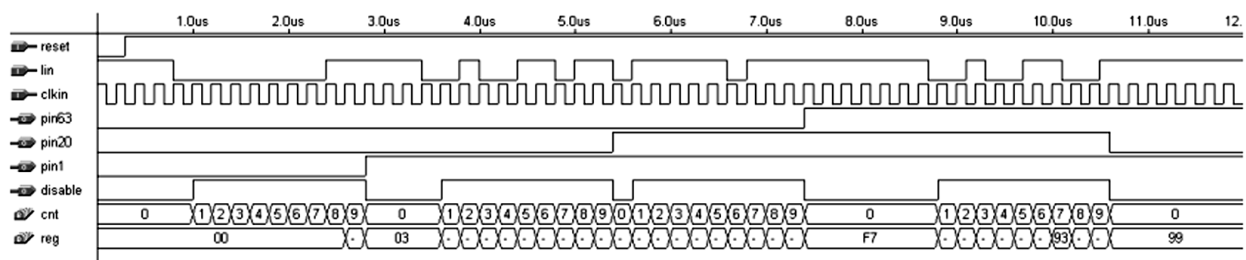


Fig. 3. Waveform of the Serial Input Shift-In.

is remedied by simply grounding out the disable pin in the hands-on version rendering both designs identical.

In order to completely avoid reducing resources, another FPGA is used to generate all the pin values. All the I/O pins from this FPGA connect to the FPGA that is to be programmed with the user's file. The I/O FPGA is programmed with EEPROMs. This entails that the FPGA to be programmed by the user has none of its resources taken up by the i/o system.

Another alternative is to slow the system clock down to a frequency where all inputs can be inputted per cycle. This requires that the input codes be shifted in at a much higher frequency than the system clock. This system was not developed, but provides an alternative solution.

Equally, the interface is connected to other important pins such as 'Done' allowing the user to view when programming is complete.

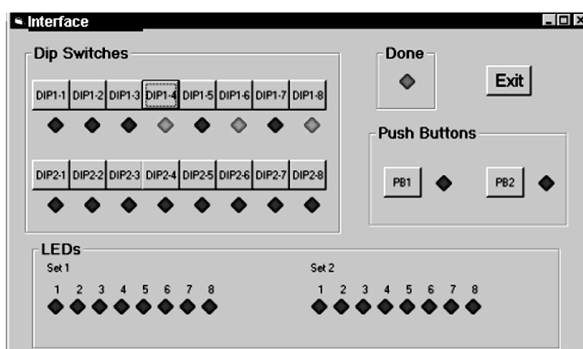


Fig. 4. Interface Software for Controlling the Inputs to, and for Detecting the Outputs from the FPGA.

The designed interface is shown in Figure 4. In the prototype interface software, 18 input pins and 16 output pins are used. There is also the 'Done' green light that indicates when the FPGA has been programmed. More functionality can be ascribed to this interface software as is needed.

#### D. Limitations

One important problem that befalls all of this system is the lack of ability to check voltages at other points than the actual probed pins throughout the FPGA. For instance, users cannot determine whether the power supply is faulty and does not supply enough voltage to the FPGA. Therefore, no oscilloscope readings are available. Thus, the user is not entirely able to detect whether it is his circuit that is not working, or the FPGA board. Furthermore, there is the issue of the parallel/serial port interface between the FPGA and the computer that are application-specific. Thus, a more sophisticated set-up would be required to allow for all applications.

As both VNC and NetMeeting are real-time programs, they both consume bandwidth. As bandwidth is costly, concessions would need to be made to allow for different restrictions, such as connection speeds, bandwidth allocations, and so on. One important consideration in the use of COTS is that the hardware and software is ever improving. As such, the bandwidth demands will eventually be met, and the system will be practical in this sense as developments are made.

#### 7. Support for Grading

Parallel to the multi-user distance laboratory (MUDL) software, we use a Web-based evaluation tool. The goal of the tool is to collect useful data so that each student's overall teamwork performance can be better addressed. Apart from evaluating students' individual proficiency on the subject, attention will also be placed on students' participation in working in a team. Currently, a web-based query-form to collect *peer evaluations* and comments from students is fully developed. The

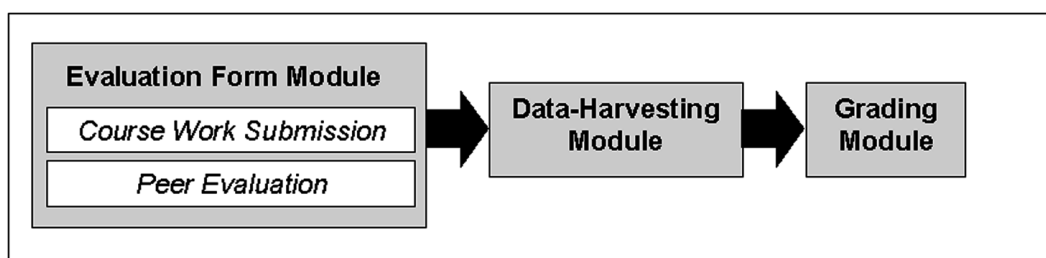


Fig. 5. Hierarchy of Grading Support System.



prospective tool will later extend to incorporate the web-based *evaluation form* for data acquisition, which is built alongside a *data-harvesting program* for data processing. The tool structure is outlined in Figure 5. The modular *grading program* component can be customized for a specific course.

### A. Peer Evaluation

Besides individual performance, interpersonal relationships are an important ingredient for effective teamwork. Participation of all team members in a project collaboration is paramount, as it builds trust among team members. Equally, through interaction, potential problems such as poorly defined expectations among team members can be eliminated. Thus, interpersonal relationship should also be accounted for during the evaluation process of team projects. However, it is traditionally difficult for instructors to understand and assess the *intra-team dynamics* [11].

The most effective way to evaluate interpersonal relationships is through *peer evaluation*, where students have the opportunity to criticize each other's performance. Students are better motivated when peer evaluation is announced beforehand. It will promote the importance of an individual's performance as well as integrity in completing their assigned tasks among other team members. We found that the tool existence alone reduced the conflicting situations in teams to well under half the cases per semester.

### B. The Evaluation Tool

The evaluation forms shown in Figure 6 gather *peer evaluations*, course work, and auxiliary information submitted by students throughout the course. After proper organization, the instructor can easily monitor progress, review submissions, place comments and assign grades. The results will be re-distributed to the students appropriately. When the tool is fully completed, data organizations, re-distributions and the repetitive task of recording scores can also be automated. Currently, the evaluation tool is used in a course Web page at [www.ece.mcgill.ca/~ee426](http://www.ece.mcgill.ca/~ee426).

### C. Evaluation Tool At Work

In grading group projects, the evaluation tool first requests each team member to state his/her assumed tasks, as well as those expected from

The screenshot shows a web browser window titled "Evaluation Page - Microsoft Internet Explorer". The page content includes a title "Peer Evaluation Page" and a welcome message: "Welcome to the microprocessor systems peer evaluation page. The purpose of this page is to provide a more convenient way for you to express your opinion. You are encouraged to comment on any students in the class. All submissions will be kept confidential. Urgent messages should be forwarded to the course instructor directly." Below this is a note: "Before completing the form, please be sure that you are responsible for the evaluation submitted. Anonymous evaluations are not accepted. Also be sure that you provide correct and adequate information to identify yourself and others. If you need assistance, simply click on one of the links on the left, information you have already entered will not be erased when you do <back>." The form is divided into two main sections. The "General Information" section contains three text input fields: "Your Name:", "Your Student ID Number:", and "Your E-mail Address:". Below these is a dropdown menu for "You are from the:" with "Thursday session" selected. The "Evaluation of your peers" section contains a "Name:" text input field and a "Comments:" text area with the placeholder text "Please type your comments here.". At the bottom of the form is another "Name:" text input field. On the left side of the page, there is a vertical sidebar with three buttons: "Course Main Page", "Classlist", and "Help".

Fig. 6. Screen Shot of Peer Evaluation Forms.

other teammates, at an early stage of the project. Extra comments will continuously be accepted as the project develops. Finally, the completed project is submitted to the tool. The submitted data will be arranged and compared, either manually or aided by the grading program. Graded are student's individual performance and his/her ability to work with other members of the team. With this information available, the instructor can make a conclusion as to the student's overall performance.

## 8. Conclusions and Future Work

In this paper, we presented the system for a Multi-User Distance Laboratory, that we developed and implemented using COTS components. Furthermore, relevant administrative issues concerning distance education were shown to benefit from our system.

In order to achieve optimal results, both VNC and NetMeeting are used in MUDL. VNC is used for ease of FPGA-to-remote user interfacing, whereas NetMeeting is used for its high quality video-conferencing abilities, along with FTP and chat. A simple interface software was designed for circuits requiring less than 18 inputs and 16 outputs, such as supplied on the Altera UP1 board. Another, more advanced

system was devised to allow for devices with greater I/O requirements, but was not employed with the current hardware.

This system provides an adequate backdrop for the development and set-up of a remote multi-user FPGA environment. Finally, Web-based tools are incorporated to the system, including those for effective *peer evaluation*.

In the future, this prototype system could be used to develop other courseware, whereby work in laboratory-based courses can be remotely performed maintaining the educational relevance of the course and exposing students to current technologies.

## References

- [1] S. HONG, ET AL., Conducting Laboratory Experiments over the Internet, *IEEE Transactions on Education*, Vol. 42, No. 3, pp. 180–185, Aug. 1999.
- [2] P. ARPAIA ET AL., “A Measurement Laboratory on Geographic Network for Remote Test Experiments”, *Proceedings of IEEE International Conference on Instrumentation and Measurement Technology, IMTC/98*, IEEE Vol. 1, pp. 206–209, 1998.
- [3] D.W. CALKIN ET AL., Visualization, Simulation, and Control of a Robotic System using Internet Technology, *Advanced Motion Control*, pp. 399–404, 1998.
- [4] J.W. OVERSTREET AND A. TZES, Internet-Based Client/Server Virtual Instrument Designs for Real-Time Remote-Access Control Engineering Laboratory, *Proc. Of American Control Conference*, pp. 1472–1476, 1999.
- [5] C.A. ELLIS, S.J. GIBBS, AND G.L. REIN, Groupware: Some Issues and Experiences, *Comm. ACM*, pp. 38–58, Jan. 1991.
- [6] K.L. MILLS, Introduction to the Electronic Symposium on Computer-Supported Cooperative Work, *ACM Computing Surveys*, Vol. 31, No. 2, June 1999.
- [7] J. GRUDIN, Why CSCW Applications Fail: Problems in the Design and Evaluation of Organizational Interfaces, *Proc. CSCW*, pp. 85–93, Sept. 1988.
- [8] MICROSOFT CORPORATION, <http://msdn.microsoft.com/developer/sdk/netmeeting/default.htm>.
- [9] AT&T LABORATORIES, Cambridge, VNC Home Page, <http://www.uk.research.att.com/vnc/>, UK, 1999.
- [10] ALTERA CORPORATION, <http://www.altera.com>.
- [11] J.W. LOCKWOOD, Automated Team Project Management and Evaluation Through Interactive Web Modules, *Proc. of IEEE International Conference on Microelectronic Systems Education, MSE 99*, pp. 45–49, July 1999.
- [12] S. MCCracken, Z. ZILIC AND H. CHAN, “Real Laboratories in Distance Education”, *Proc. Int. Conf. Adv. Infrastructure for Business, Science and Education on Internet, SSRR2000*, L’Aquila, Italy, Aug. 2000.
- [13] XILINX, INC., *The Programmable Logic Data Book 2000*.
- [14] E. DEMKO, Commercial-Off-The-Shelf (COTS) – A challenge to military equipment reliability, In *Annual Reliability and Maintainability Symposium*, Las Vegas, NV, pp. 7–12, Jan 1996.
- [15] D. BOLAND, ET AL., Calibration of a COTS Integration Cost Model Using Local Project Data, In *Proceedings of the 22nd Annual Software Engineering Workshop*, pp. 81–98, Dec 1997.
- [16] JEFFREY VOAS, COTS Software: The Economical Choice, *IEEE Software*, 15(2):16–19, Mar. 1998.
- [17] MICROSOFT CORPORATION, <http://www.microsoft.com/windowsxp/pro/using/howto/gomobile/remotedesktop/default.asp>
- [18] R. IQBAL, A. JAMES AND R. GATWARD, “A Framework for Integration of CSCW”, *Proceedings of 7th International Conference on Computer Supported Cooperative Work in Design*, pp. 43–48, 2002.

Received: August, 2002  
Revised: March, 2003  
Accepted: March, 2003

Contact address:

Stuart McCracken  
Zeljko Zilic  
Hoi Yun Henry Chan  
McGill University  
Montréal, Canada

e-mail: {smc|cra|zeljko|henry}@macs.ece.mcgill.ca

---

STUART MCCracken received his M.Eng. and B.Eng. degrees from McGill University in 2002 and 2000, respectively. He is currently working for Analog Devices in Boston. His interests are in FPGA testing and applications.

---



---

ZELJKO ZILIC is Assistant Professor at McGill University, currently holding Chercheur Stratégique research chair. He received his Ph.D. and M.Sc. degrees from the University of Toronto and Dipl.Ing. degree from the University of Zagreb. His interests are in the design, test and verification of systems on a chip.

---



---

HOI YUN HENRY CHAN is a Ph.D. student at McGill University, from which he received his M.Eng. and B.Eng. degrees. He and his supervisor Z. Zilic received Myril B. Reed Best Paper Award from IEEE International Midwest Symposium on Circuits and Systems in 2001 for their work on substrate noise modeling, which is the topic of Henry’s Ph.D. thesis in progress. His other interests include photography and music playing.

---