Client-Server Architecture for Remote Experimentation for Embedded Systems

MJ.Callaghan, J.Harkin, TM.McGinnity and LP.Maguire
Intelligent Systems Engineering Laboratory,
University of Ulster, Derry, Northern Ireland

***Abstract*— The proliferation of web based distance education courses in recent years poses unique challenges for the teaching of disciplines involving a high level of practical work. Increasingly web based engineering courses are on offer, augmented by the provision of remote experimentation laboratories. The degree of functionality and level of user access to these remote laboratories has evolved in recent years expedited by advances in web applications and technologies. However the design and implementation of effective and usable remote experimentation facilities poses unique challenges given the inherent complexities of the learning environment and the constraints imposed by a web based delivery mechanism. In addition an overriding challenge in the implementation of remote experimentation environments is to maximise the use of existing resources without incurring substantial redevelopment costs needed to adapt hardware and software tools with the extra functionality needed for online remote access. This paper presents and demonstrates a client-server architecture for distant access to an integrated learning environment for remote experimentation for embedded systems which compliments and augments current web based e-learning approaches while fully utilizing existing on-campus resources without substantial redevelopment effort.**

***Index Terms*— E-learning, Embedded Systems, Remote experimentation.**

I. INTRODUCTION

The proliferation of web based distance education courses in recent years poses unique challenges for the teaching of disciplines involving a high level of practical work. Experience in these areas have shown that a complementary interdependent approach, combining theoretical material underpinned by practical exercises, is essential for effective learning [2]. Increasingly electronic and electrical engineering related distance education courses are offered online. Constant innovation and product evolution, particularly in the area of embedded systems, necessitates educational institutions and other training providers to continually reassess the content and delivery of engineering curricula making it necessary to devise, implement and evaluate innovative pedagogical approaches to teaching without compromising the cultivation of traditional skills [1].

Remote experimentation facilities offered as part of a web-based learning approach affords a number of critical benefits, augmenting and complementing web based learning material by facilitating distant access to campus based physical resources [3, 4]. For engineering related distance education courses the use of a web based delivery mechanism is the only realistic method of providing practical hands-on experience, allowing remotely located students to complete laboratory assignments, unconstrained by time or geographical considerations while developing skills in the use of real systems and instrumentation. Remote experimentation laboratories have evolved in recent years, expedited by advances in web applications and related technologies, where the end objective is to accurately recreate the on-campus laboratory based student experience with an equivalent level of user access, functionality and flexibility. However the design and implementation of effective and usable remote experimentation facilities poses unique challenges given the inherent complexities of the learning environment necessitated by the need for the cohesive integration of a diverse range of complex components and further hindered by the constraints imposed by a web based delivery mechanism.

This paper addresses this issue in the context of web based learning resources for embedded systems related courses. An integrated learning environment is presented, augmented by advanced remote experimentation facilities, which successfully achieves the necessary complimentary balance of theoretical material delivery and practical experience offering the student a more complete learning experience. The approach used is based around a client-server software architecture and includes a straightforward solution to the reuse of existing hardware and software resources in this context. Section 2 of this paper summarises recent research in the area of remote experimentation for embedded systems, section 3 introduces the Distance Internet-Based Embedded System Experimental Laboratory (DIESEL) project, lists the project objectives and presents an integrated learning environment based on a client-server approach for remote experimentation for embedded systems. Section 4 of the paper describes the individual components of the environment. Section 5 of this paper summarizes recent research, discusses outstanding issues and future directions for further investigation.

II. REMOTE ACCESS LABORATORIES

This section reviews current remote experimentation projects and identifies the focus and functionality of each project and the overall deficiencies of the current approaches. The key features required for the implementation of effective, functional and usable remote experimentation laboratories are highlighted. Several remote access laboratories focusing on areas as diverse as science and engineering are now accessible online.

The University of Singapore provides a virtual laboratory focused on experiments related to RF communications [4], the Cybernetics Laboratory Norway [5] allows students to investigate and control industrial refrigeration processes using different control algorithms, the University of Texas at Arlington provides similar facilities to allow students to investigate issues related to the classical Inverted Pendulum control problem [5]. The University of Chattanooga [6] offers online experiments in process control systems and dynamics, allowing users to control a number of parameters including pressure, temperature and voltage. Weblab at the Illinois Institute of Technology is a collection of science and engineering Web-based experiments designed to allow remote control of real "hands-on" devices and equipment [7]. AIM [8] is a collaborative project between the Rensselaer Polytechnic Institute and the Norwegian University of Science and Technology where the remote laboratory is used to complement existing courses in semiconductor technology and device characterisation. The PEARL project (Practical Experimentation by Accessible Remote Learning project) is aimed at the development of systems enabling students to conduct real-world manufacturing engineering and digital electronic experiments [9]. The University of Zagreb provides several virtual laboratories allowing access to microcomputer development boards, dc motors and a monitor/debugger program for courses involving embedded systems [10].

Other notable remote access laboratories include Gonzalez-Castona [11], the University of Oulu, Finland [12]. These projects generally focus on microprocessor based experiments for postgraduate and undergraduate computing degrees. More recently, non-academic institutes such as Techonline [13] developed Virtualab focusing on embedded systems training for the continuing professional development market. In a similar approach, the Relax project [14] investigates the possibility of developing a new business model where high quality remote experiments and support material can be made available to a worldwide market. A more comprehensive, in-depth review of current online remote access laboratories is available here [15].

From this review process a number of key deficiencies in existing remote experimentation laboratories were identified. It was concluded that in contrast to traditional laboratories, web based remote access facilities are crude in nature with only a fraction of the functionality, accessibility and flexibility of their campus based counterparts and failed to fully utilize existing hardware and software resources [16]. To address these deficiencies, key features and functionality currently available in campus based laboratories were identified that would need to be replicated in remote access facilities to make the overall experience comparable. These features include facilitating full functional access and remote control of a diverse range of software and hardware resources. In the context of electronic engineering and embedded systems this would include facilitating remote access to software tools e.g. cross compilers and VHDL synthesis tools to allow the user to program modern experimental equipment e.g. Microcontrollers, Systems on Chip (SoC) and real-time operating systems (RTOS) while allowing the remote control of debugging instrumentation e.g. logic analysers and oscilloscopes. Full functional access and remote control in this context would be defined as allowing the student to remotely build, debug and test circuits, connect instrumentation to test points on experimental boards as required and have full access and control to all the functionality of the test instrumentation using a web based platform while overcoming all the inherent constraints that the use of this delivery medium entails. The next section introduces and describes the DIESEL project (Distance Internet-Based Embedded System Experimental Laboratory) which is an integrated learning environment for remote experimentation which seeks to address many of the issues raised in this review.

III. CLIENT-SERVER ARCHITECTURE FOR REMOTE EXPERIMENTATION

This section catalogues our recent experiences in the practical implementation of the DIESEL project (Distance Internet-Based Embedded System Experimental Laboratory) a three year distance learning project funded by the EPSRC under the Masters Training Program and located at the Intelligent Systems Engineering Laboratory on the Magee campus of the University of Ulster, Northern Ireland [17]. The rationale for the DIESEL system development is outlined, the range of experiments offered catalogued, the system architecture and components of the integrated environment are described in detail and a typical user session demonstrated.

The overall objective of the DIESEL project was to develop remote-access laboratories for Embedded Systems modules on a range of undergraduate and postgraduate courses. Traditionally students attended practicals in campus based laboratories at fixed times during the academic year. This approach restricted access to laboratories resources to traditional working hours which did not meet the needs of students requiring more flexible attendance patterns to meet current lifestyle commitments.

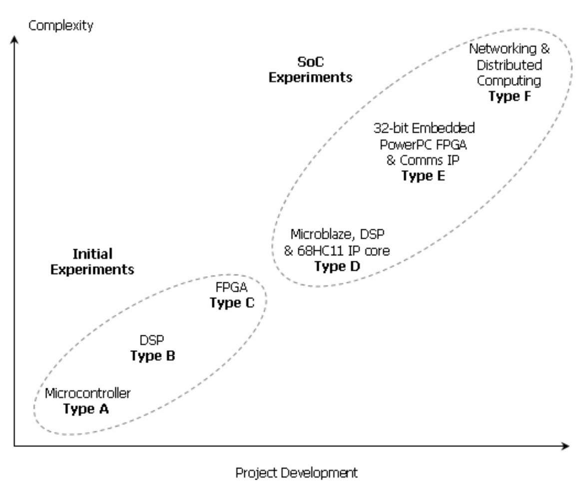

Figure 1. Experimental roadmap

Six different classes of experiments were chosen (figure 1) exposing the students to a comprehensive in-depth range of modern embedded system technologies and design tools (figure 2). The experiments range in complexity from Microcontroller and Digital Signal

Processing (DSP) experiments to Field Programmable Gate Array (FPGA) design, Systems on a Chip (SoC) and Real Time Operating Systems (RTOS). Within each of these classes a range of individual experiments are offered. This broad functional remit covers a diverse range of experimental boards and test equipment which increases the overall complexity of the implemented architecture and resulting environment but was necessary to facilitate a high degree of system flexibility and functionality.

Experiment Type	Description	Equipment
A	Microcontroller based experiments on I/O signalling	Motorola 68HC11 development platform & IDE
B	DSP based FIR filtering	Texas TMS320C50 DSP development platform & IDE
C	VHDL State machine design and verification in FPGAs	Xilinx Virtex-II development board. VHDL Synthesis tools
D	RISC processor implementation & System on a Chip (SoC) Design	Motorola 68HC11, DSP & Xilinx Microblaze IP cores. Moderate sized Xilinx Virtex-II development platform
E	CISC processor and System I/O development in SoC environment	Xilinx Embedded PowerPc FPGA development platform & Communication (UART, PCI) IP cores.
F	Real-time OS development, networking and distributed computing	Windriver Vxworks Real-time OS and Tornado IDE, TCP/IP stack protocol

Figure 2. Experiment classification, description and technology

A. *DIESEL client/server architecture*

The overriding challenge in designing this complex environment was to fully utilize existing resources without incurring substantial effort and redevelopment costs in the redesign and adaptation of existing hardware and software tools with the extra functionality needed for online remote access, a costly undertaking which would be beyond the expertise and budget of most teaching institutions and training providers. To address this challenge a generic architecture and access-control methodology for remote laboratories was developed.

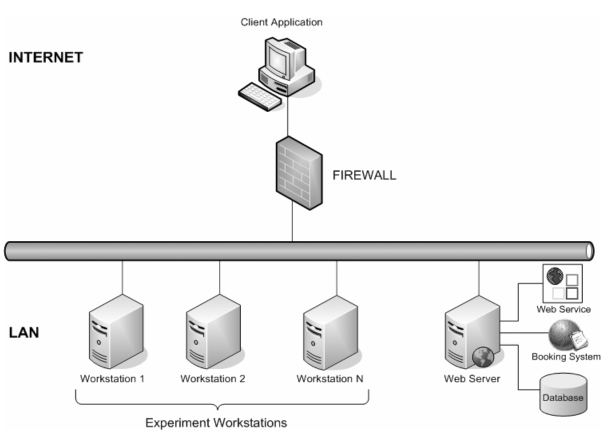

Figure 3. Architecture of remote access laboratory

This architecture efficiently integrates instrumentation and experimental hardware components while allowing full access and remote control of any software components and consists of a gateway server (laboratory administrator) connected to the Internet and a number of experimental workstations connected to the server via a network-hub (figure 3).

Students accessing the remote laboratory initially connect to the gateway server which handles administration and authorization duties and connects validated users to available experimental workstations. Each individual workstation is identical and hosts a range of experimental related hardware and software tools required to carry out practical experiments. Figures 4 and 5 illustrate the hardware components of a workstation which includes test instrumentation and a range of experimental boards. The test instruments are configured and controlled from the workstation using the GPIB protocol while the experimental boards are accessed from the workstation using either RS232 or parallel connections as required. A GPIB controlled switching matrix allows the test instrumentation to be connected to a number of test points on the experimental boards as necessary in the course of an experiment. Each workstation also contains an extensive range of software tools necessary for the design, editing and compiling of embedded software programs.

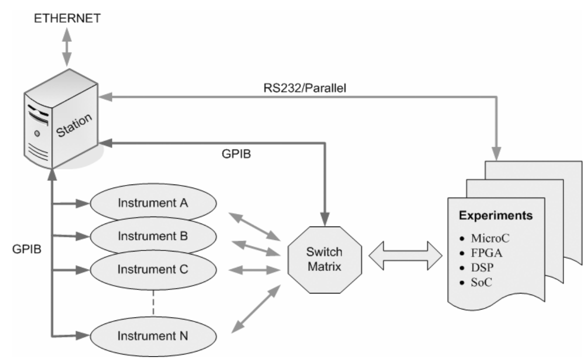

Figure 4. Generic DIESEL architecture

To enable these hardware and software resources to be remotely accessed and controlled the DIESEL software has been developed [10]. The DIESEL software uses a distributed software architecture developed using Microsoft .NET technology. The architecture for the remote access laboratory consists of a client application, a number of experimental workstations and a web server which hosts a database, a web service and a web-based booking system. Figure 6 shows the shows the communication and data flows between the various parts of the remote access lab. Communication between the client application and the server application on the experimental workstation uses direct peer-to-peer TCP communication and operates over a secure encrypted .NET Remoting channel using 256-Bit encryption. This ensures the safe delivery of commands and data between the client application and the server application. Interaction with the web service occurs over HTTP and messages and data are exchanged using the Simple Object Access Protocol (SOAP). This approach circumvents any problems that could arise with access through firewalls and avoids cross platform issues [10]. The system implements a four-tier communication model: the presentation layer, the data layer, the business layer and the physical layer (figure 7).

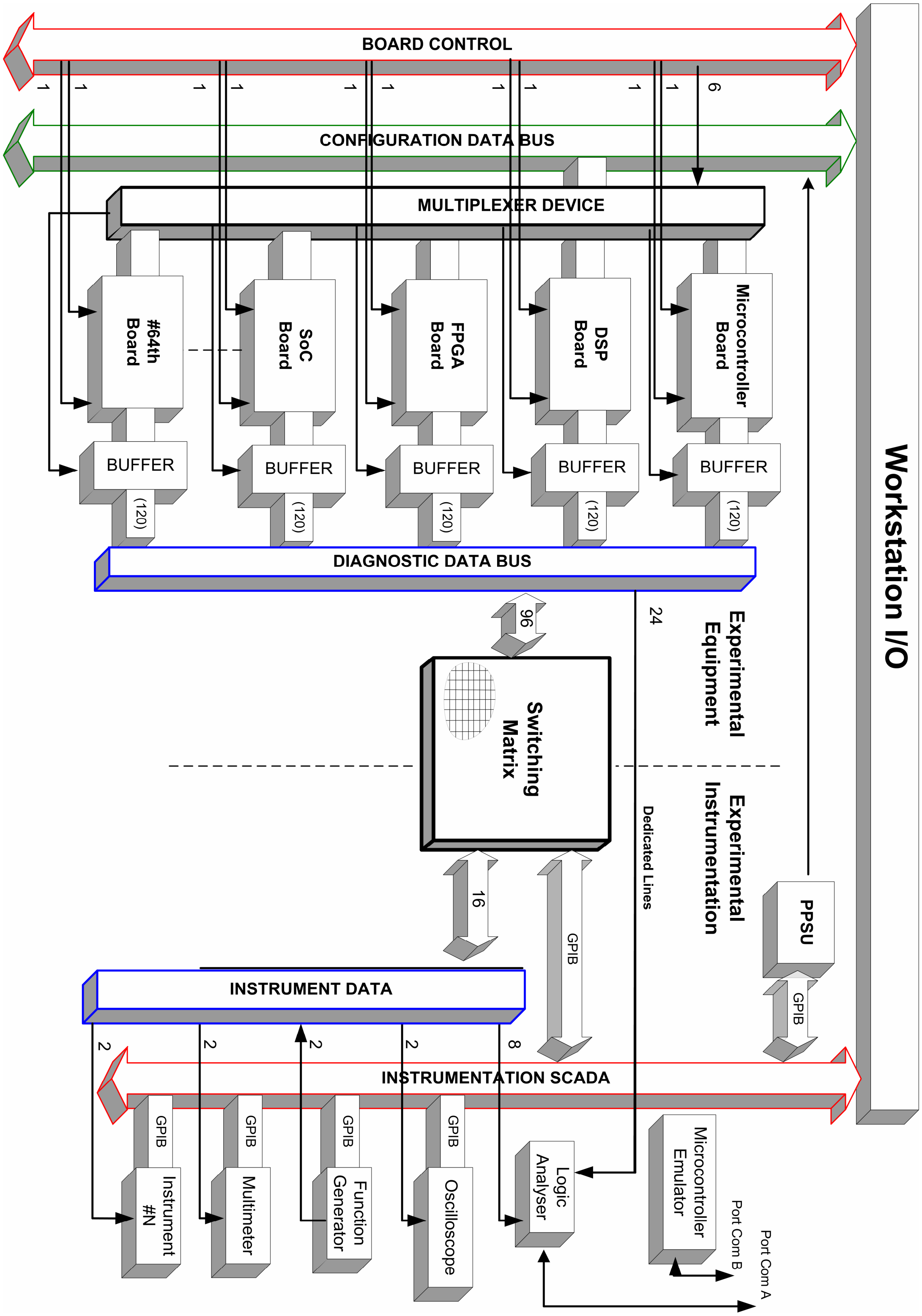

Figure 5. DIESEL hardware architecture

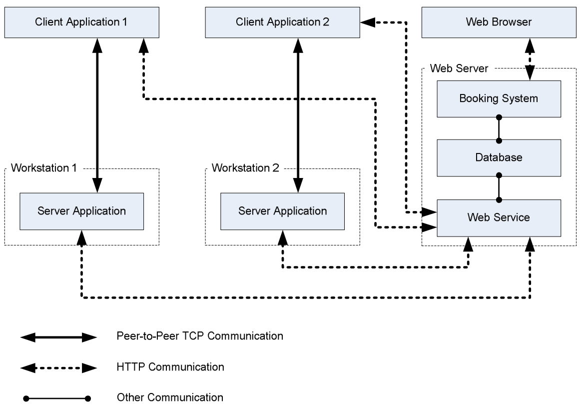

Figure 6. Communications protocol in DIESEL system

The presentation layer consists of the DIESEL client application and the booking system which is accessed through a web browser. The DIESEL client provides the user interface which allows the user to configure and manipulate embedded systems and instruments remotely. The booking system provides the user interface for making and managing bookings. The data layer provides access to the database through either a web service or the web booking system. The business layer is implemented in the DIESEL server application, and provides access and control to the physical layer. The physical layer consists of all of the hardware resources (e.g. experimental boards and test instruments).

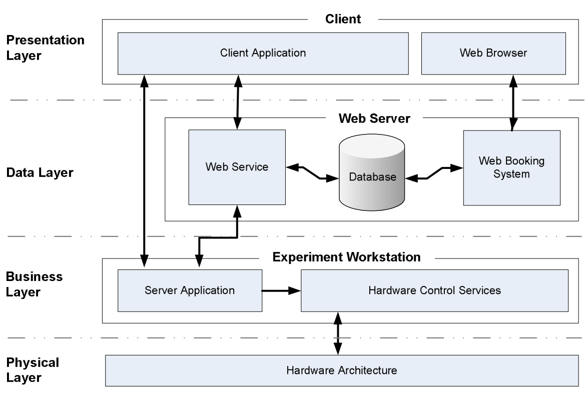

Figure 7. DIESEL four tier communication model

A web service is used as a gateway between the presentation, business and data layers to allow the client application and server application to access the database. This approach was preferred as it allows the separation of the client and server applications from the data storage process. The server application (business layer) responds to commands from the client application by executing the appropriate control programs on the hardware architecture (physical layer) to configure the embedded circuits, signal routers and instruments while sending commands to the circuit under test.

B. *Accessing the remote laboratory*

Access to the remote laboratory is done through the DIESEL Client software which must first be installed on the client computer. Before a remote experimentation session can be started a booking must be made. This is done through the online booking system on the DIESEL website (figure 8). When a session has been booked the user can launch the DIESEL client application which will present a login screen.

Figure 8. Remote environment booking system/DIESEL client login

Figure 9 shows the typical sequence of interactions between the various components of the DIESEL software during application initialization and remote experimentation.

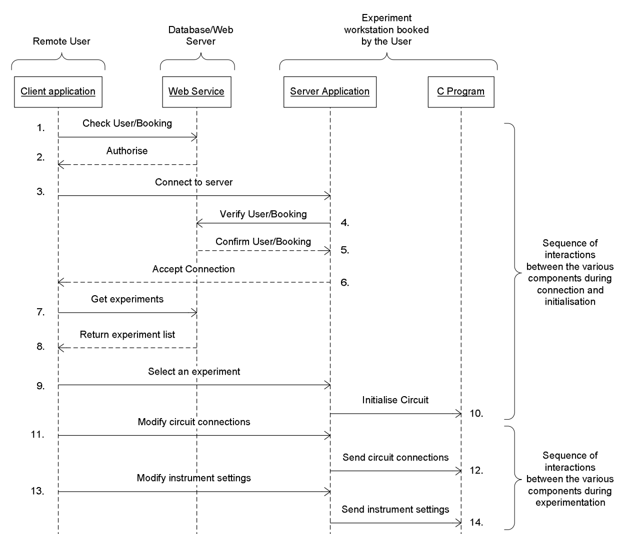

Figure 9. DIESEL Client/Server operation

During the initialisation stages (1–10) the DIESEL client attempts to connect to the remote experiment workstation. The client application sends the username and password to the web service, which verifies the user's credentials against the database and checks if there is a valid booking for the current time. If there is a valid booking the web service will return the IP address of the experiment workstation to which the user is booked and grants authorisation to the client application to continue connecting to the remote experiment station. The client application then sends a connect request to the server application, again passing the username and password.

The server application will confirm the user's authorisation and booking with the web service before accepting the connection. When connected the client application will retrieve a list of available experiments from the web service and the user will be presented with an experiment choice dialog window (figure 10).

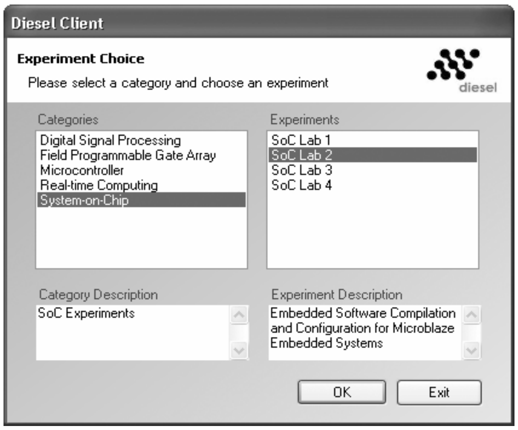

Figure 10. Experiment choice dialog window

After the user has selected the experiment that they wish to carry out the client application will begin to retrieve all of the relevant support material (e.g. experiment instructions and lecture video). The application will also load the virtual circuit and instrumentation and will send a message to the server to initialise the circuit. The server will pass on the message on to a control program which will tell the switching matrix to connect the chosen experiment to the instrumentation. The remaining interactions (11–14) involve the user working on the experiment. As the user changes the circuit connections or instrumentation settings the commands are sent back to the server which configures the equipment and switches between instruments and development boards as required. This section of the paper introduced the DIESEL client-server architecture and discussed the overall operation of the system. Section 4 of the paper describes individual components of the architecture and their interaction in the context of an integrated learning environment for remote experimentation.

IV. DIESEL INTEGRATED LEARNING ENVIRONMENT

The DIESEL client application is composed of components which make up the DIESEL integrated learning environment and is comprised of two complementary sections, learning support resources and remote experimentation/access facilities (figure 11). The DIESEL software environment provides a hierarchical tabbed navigation system so that users can quickly navigate through the various features and resources provided by the application (Figure 12).

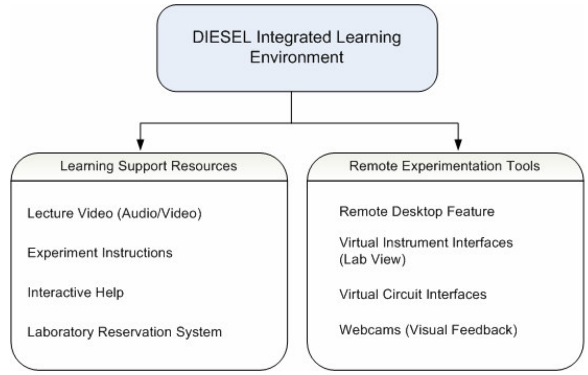

Figure 11. DIESEL Integrated learning environment

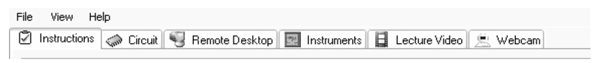

Figure 12. DIESEL environment tabbed navigation system

A. Learning Support Resources

The learning support resources provide the user with the basic assistance needed to begin conducting remote experiments. The resources available to the user include extensive lecture notes and experiment instructions as well as audio/video content providing relevant theoretical background material and guidance for the experiments. A comprehensive help system is available outlining and demonstrating general operation of the remote laboratory and the conduction of experiments. The system also provides the facility to schedule an experimentation session through the web-based laboratory reservation system.

B. Remote Experimentation Facilities

The remote experimentation facilities are central to the overall viability and functionality of the DIESEL environment and contain all the features necessary to allow students to effectively recreate the campus based laboratory experience remotely. The remote desktop, virtual instrumentation, virtual circuit, and live experiment board webcams provide the user with genuine control and visualization of real laboratory equipment. In recent years remote desktop access packages e.g. Microsoft Remote Desktop, have become increasingly commonplace [19]. These packages allow users to access remotely based machines via the Internet from any location through either a windows-based client or a web browser. When granted access to a remote machine the user can then use all the applications, files and software resources on the workstation with the same level of functionally as if physically present in the same location. Utilizing this approach in the context of remote experimentation applications offers a simple but very effective solution to one of the main deficiencies highlighted earlier, namely the ability to fully utilize existing software, hardware and test equipment in a remote laboratory environment while avoiding large redevelopment efforts with minimum costs incurred. This thin client approach facilitates full access to existing software resources remotely, without adaptation and relieves the user of the necessity to install or configure complex and expensive software tools on their home machines. In addition all compilation and processing operations are carried out remotely freeing up local computational resources. This approach also has important implications as it allows the use of more complicated test equipment over the Internet through the DIESEL client e.g. logic analysers with desktop based control software. Figure 13 shows the DIESEL client with remote desktop facility and clearly demonstrates the effectiveness of this approach using an example logic analyser software controlled program.

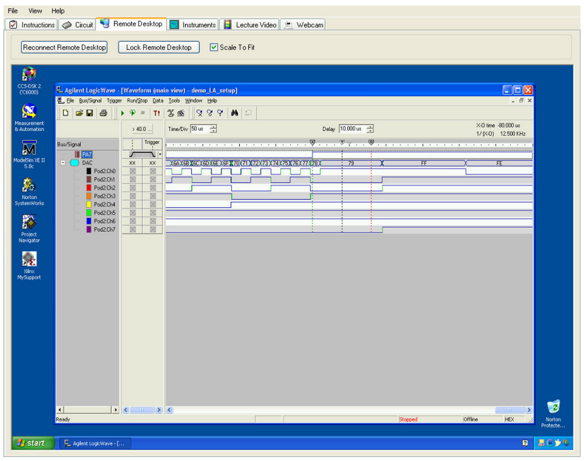

Figure 13. DIESEL Remote Desktop approach with Logic Analyzer

C. Virtual Instrument interfaces

A major objective of this project was to recreate, as closely as possible, the experience of conducting a practical campus based engineering laboratory for the remote student. Achieving this goal would allow the student to acquire a number of key skills which included familiarity with setting up and configuring laboratory test equipment, the wiring up and troubleshooting of test circuits and the taking and recording of accurate readings and experimental data. In terms of instrumentation control it was desirable that the interfaces developed to access and control the remote equipment replicated as closely as possible the physical layout of the actual instruments while facilitating the highest level of functionality. This was important as it would allow the student to move seamlessly from virtual representations of test equipment in a remote laboratory environment to actual test equipment in a real laboratory and to be capable of using this equipment competently as a direct result of their online experiences. Labview [20] was used to achieve this as it allows the efficient creation of customised functional user interfaces. Virtual interfaces for three pieces of instrumentation were developed, an oscilloscope, a function generator and a digital multi-meter (figure 14). Further development was required to fully integrate the Labview interface with the backend control/switching system as it was necessary to restrict instrumentation output values on an individual experimental basis.

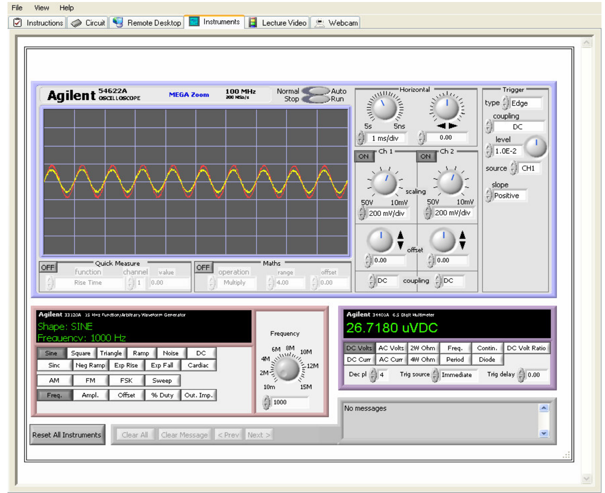

Figure 14. Virtual instrumentation in Labview

D. Virtual Circuit Interfaces

After developing the instrument interfaces the next area investigated focused on how the remote student could setup an experiment, wire up test circuits and move instrument probes around different circuit analysis points. This problem broke down into two subtasks, how to make dynamic physical connections between the inputs and outputs of different test instruments and the circuit analysis points and secondly how to allow the user to make these connections remotely.

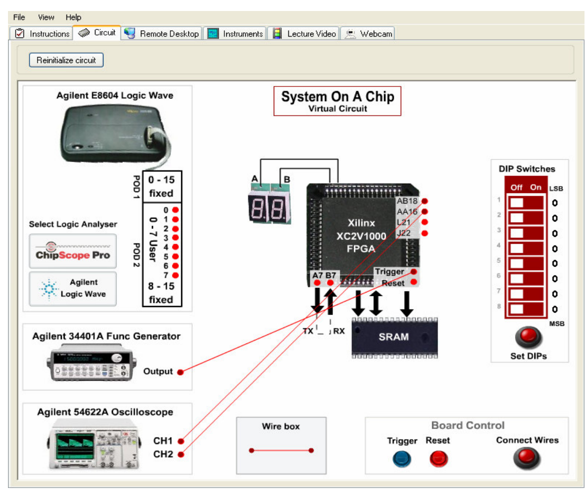

Figure 15. Virtual circuits implemented in Macromedia Flash

The first task was reasonably straightforward and solved using a GPIB protocol based Pickering switching matrix (Figure 4). Connections to all valid circuit test points were hard wired through the matrix as was the connections to the instruments. Labview was initially used as a front end to the switching matrix allowing the user to configure connections between test points as required. However following initial testing it was felt that while the Labview interface was functional it lacked flexibility and did not recreate the experience of physically wiring up/ debugging a circuit.

After further investigation in which a number of interface options were investigated it was eventually decided to use MacroMedia Flash [21] as the front end and user interface to the switching matrix. The Macromedia Flash interfaces developed represent the physical layout of the circuit under test and the test equipment (Figure 14). A number of hotspots are present in the interface representing valid connection points between test equipment and circuit analysis points. To make a circuit connection the user “drags” the connector from the equipment i.e. channel 1 of the logic analyzer and places it on desired circuit analysis point (an event). The connector “locks on” to this point and will stay connected until moved again. At the backend this action causes the desired physical connection to be made through the switching matrix. This approach allows the remote user to effectively setup a test circuit and wire it together. The interface is flexible enough to allow the user to trouble shoot and debug a circuit but safeguards have been included in the programs to disable inappropriate circuit connections.

E. Visual feedback

To provide visual feedback to the students during experiments where necessary, a series of Web cams (figure 15) have being integrated into the client using the Macromedia Flash Communication server [21]. The Flash Communication server allows users to create web based interactive media applications combined with video streaming capabilities.

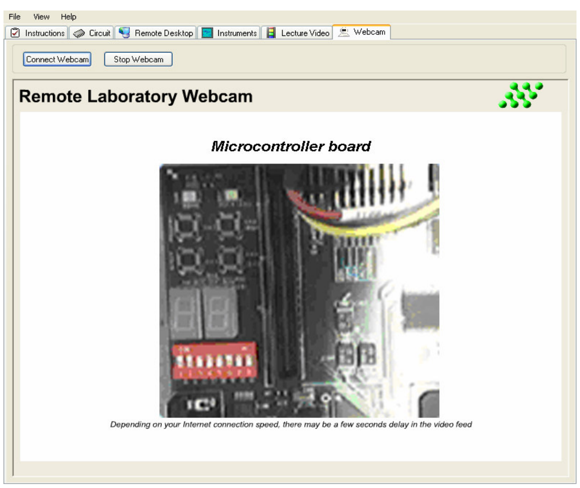

Figure 16. Visual feedback

This section of the paper presented the components of an integrated learning environment for remote experimentation. A structured environment for remote experimentation was introduced which included comprehensive teaching and support material and integrated the ability to complete advanced embedded system experiments remotely.

F. Testing and evaluation

This environment has undergone extensive testing and was deployed on a number of Undergraduate and Postgraduate courses. In total over 40 students used the environment over a semester. General feedback related to the downloading and installation of the client software on the remote user’s machine highlighted the ease of installation and configuration. Comments related to the user interface and help features were positive but the issue of user support and help at unsociable hours was raised. The flexibility of 24 hour access to campus based resources was welcomed and some additional client features requested (integrated booking system). Further information on the development and deployment aspects of this project is available here [15, 16].

V. DISCUSSION AND CONCLUSION

A review and discussion of recent work undertaken by the authors in addressing issues related to the implementation of a remote access experimental laboratory for embedded systems has being presented. A client-server architecture implemented using a web services/.NET framework for remote access was described and the individual components of the integrated learning environment presented. This approach affords a number of critical benefits allowing remotely located students to complete laboratory assignments unconstrained by time or geographical considerations while facilitating the development of skills in the use of real systems and instrumentation. In addition the approach taken in designing the overall environment allows extensive use of existing hardware and software resources without necessitating substantial redevelopment and redesign effort. This integrated learning environment, while initially developed for educational use, has great potential for use in the continuing professional development market. Future work in this area will concentrate on extending the range of experiments available in the system and on implementing help systems for the more complex experiments detailed in the experimental roadmap.

REFERENCES

[1] D.Beetner, H. Pottinger, K. Mitchell, “Laboratories teaching concepts in microcontrollers and hardware-software co-design”, 30th Annual Frontiers in Education Conference, vol.2, 2000, pp.S1/1-5

[2] W. Wolf, J. Madsen, “Embedded systems education for the future”, Proceedings of the IEEE, vol.88, no.1, 2000, pp.23-30

[3] M. Cooper, “The challenge of practical work in an eUniversity - real, virtual and remote experiments”, IST 2000 Conference, 2000, pp. 34 -40

[4] Cybernetics Laboratory. http://www.cyberlab.org/

[5] T. Fjeldly, M. Shur, H. Shen, T. Ytterdal, “Automated internet measurement laboratory (AIM/Lab) for engineering education” Frontiers in Education: 29th Annual Frontiers in Education Conference, vol.1, 1999, pp. 12A/2-3

[6] http//chem.engr.utc.edu

[7] M.Hites, 1999, "Implementing and Evaluating Web-Based "Hands-On" Laboratories For Undergraduate Education". ASEE IL/IN Sectional Conference , 201 - 209

[8] H.Shen et al., 1999, "Conducting Laboratory Experiments over the Internet", IEEE Trans. on Education, 30, 191-199

[9] http://www.open.ac.uk/

[10] G.Muzak and I.Cavrak,2000, “The virtual laboratory project”, Int. Conference Information Technology Interfaces, 241-246

[11] F.G. Gonzalez-Castono, L.Anido-Rifon, et al., 2001, “Internet access to real equipment at computer architecture laboratories using the Java/CORBA paradigm”, Computers and Education Journal (Elsevier), 36, 151-170

[12] J. Paaso and T.Manninen , 2001, "Computer supported and controlled learning with the aid of Internet technologies", Int. Continuing Engineering Education and Lifelong Learning, 11, no. 1/2, 95-106

[13] http://www.techonline.com/

[14] F. Bjarne, E.Kjell, T.Malvig, I. Eikaas, 2001 "Remote Experimentation – New content in Distance learning" International Conference on Engineering Education, August 6 – 10, 2001 Oslo, Norway

[15] MJ. Callaghan, J.Harkin, TM. McGinnity, LP. Maguire, "An Internet-based methodology for remotely accessed embedded systems", IEEE International Conference on Systems, Man and Cybernetics, October 2002

[16] MJ. Callaghan, J.Harkin, TM. McGinnity, LP. Maguire,"Integrated Architecture for Remote Experimentation", Proceedings of 2003 IEEE Int. Conf. Systems, Man and Cybernetics, Washington, DC, USA, October 2003, IEEE, Pages 4822-4827

[17] Intelligent Systems Engineering Laboratory. [Online]. Available: http://isel.infm.ulst.ac.uk/

[18] Callaghan MJ, Harkin J, McGinnity TM, Maguire LP, "Adaptive Intelligent Environment for Remote Experimentation", Proceedings of IEEE/WIC International Conference on Web Intelligence, Canada, 13-17th October 2003, Pages 409-412

[19] http://www.microsoft.com

[20] http://www.ni.com

[21] http://www.macromedia.com

AUTHORS

Michael Callaghan is a Lecturer in the School of Computing and Intelligent Systems at the University of Ulster. He is a member of Intelligent Systems Engineering Laboratory with the University of Ulster. His current research interests relate to the Remote Experimentation and Hybrid Intelligent Systems. (e-mail: mj.callaghan@ulster.ac.uk)

Jim Harkin is a Lecturer in the School of Computing and Intelligent Systems at the University of Ulster. He holds a Bachelor of Technology in Electronic Engineering, Masters of Science and Phd from the University of Ulster, and is a member of the IEE. He is a member of Intelligent Systems Engineering Laboratory with the University of Ulster and his current research interests relate to the design and implementation of intelligent reconfigurable embedded systems. (e-mail: jg.harkin@ulster.ac.uk)

Martin McGinnity is Professor of Intelligent Systems Engineering at the University of Ulster. He holds a first class honours degree in physics, and a doctorate from the University of Durham. He is a Fellow of the IEE, member of the IEEE, and a Chartered Engineer and leads the research activities of the Intelligent Systems Engineering Laboratory. His research interests relate to the creation of intelligent computational systems and the area of intelligent systems in general. (e-mail: tm.mcginnity@ulster.ac.uk)

Liam Maguire is a Reader in the School of Computing and Intelligent Systems, University of Ulster. He obtained a MEng (distinction) and a PhD in Electrical and Electronic Engineering from the Queen's University of Belfast. His current research interests relate to the creation, design and implementation of intelligent systems. (e-mail: lp.maguire@ulster.ac.uk)

Manuscript received April 21, 2005. This research was funded with the financial support of the EPSRC (UK) (Reference GR/N26753).