World Transactions on Engineering and Technology Education
Vol.3, No.1, 2004

The development of a remote laboratory: educational issues

Bruce D. Moulton, Vladimir L. Lasky & Stephen J. Murray

University of Technology, Sydney
Sydney, Australia

ABSTRACT: Engineering graduates, employer groups and professional organisations all recognise the importance of communication, team and time management skills. As educators of engineers, we try to design learning activities that facilitate the development of these attributes. A remote laboratory has been operating at the University of Technology, Sydney in Sydney, Australia, since 2001. This enables staff and students to use embedded systems hardware from remote locations. Students of the subject, *Operating Systems*, now make extensive use of the remote laboratory, which has Web-based vision and peer support. The environment enables students to control, monitor and program real embedded system components from a remote location. Modifications have been made as a result of a re-analysis of the system in the light of observations made during the semester, suggestions made by users and consultations with other staff. Questions have been asked to determine if the environment adds value to learning. Further investigation of the relationships between the learning activities and learners' technical, communication and time management competences are warranted.

LEARNING OUTCOMES

A recent survey of engineering graduates from the University of Technology, Sydney (UTS), in Sydney, Australia, who have been identified by their employers as *highly successful*, indicates that these graduates considered their most important attributes to be as follows:

- Being able to develop and contribute positively to team-based projects;
- Being willing to face and learn from errors and listen openly to feedback;
- Being able to organise work and manage time effectively;
- Understanding personal strengths and limitations [1].

The *most important attributes* identified by these graduates align closely with several of the National Generic Competency Standards of the Institute of Engineers Australia (IEAus). Indeed, the IEAus specifies that professional engineers should have attributes that include the following:

- Manages own time and own processes;
- Communicates effectively with others;
- Develops and maintains the trust and confidence of colleagues, clients and suppliers through competent performance;
- Seeks and values input from internal and external sources to enhance communication;
- Mentors others in specific areas of engineering focus;
- Builds and maintains network relationships that value and sustain a team ethic [2].

Similarly, reports by organisations such as the Department of Education, Science and Training, the Australian Industry Group and the Business Council of Australia conclude that employers are increasingly interested in the same skills, particularly communication skills.

Currently, there is increased demand for engineering graduates who can perform competently in remote development environments. Remote development is a process whereby engineers participate in projects from different geographic locations. These kinds of projects are being seen more and more, so it is important that our students gain experience in these types of environment.

The challenge is to design courses and learning activities in a way that facilitates students' development of the required attributes. In attempting to meet this challenge, we, as educators, need to focus not just on the physical and technical aspects of our educational environment, but also on the learning activities and outcomes for our students.

REMOTE LABORATORIES

In the higher education sector, there has been an increasing emphasis on flexible learning, including online learning. This has presented new opportunities, as well as challenges. At an increasing number of universities, Web-based courseware is used in teaching laboratory-based courses [3]. Such developments have led to concerns about the implications of decreased face-to-face teaching [4][5]. Of particular concern to many people has been the replacement of real laboratory sessions with laboratories that are *virtual*, *Web-based*, *online* or *remote*.

A sizable proportion of virtual laboratories employ Java applets (or JavaScript) that connect to applications like *MATLAB*/ *Simulink* and *LabVIEW*. Examples are the Politechnica University of Bucharest electrical engineering VLab [6] and the Universidad Nacional de Educacion a Distancia virtual control laboratory Java interface for *MATLAB* [7].

Simulation

Simulation has its advantages. Some systems (such as nuclear plants) are too dangerous and expensive to allow real world experimentation. However, simulation does not provide the same *feel* as working in a real laboratory. In recognition of this problem, a number of online laboratories allow users to connect to real-world laboratory devices. An example is the VLAB used by the Faculty of Engineering at the National University of Singapore, which allowed remotely located users to see and control real laboratory instruments [8]. Another example is the Automatic Control Telelab, which has been used in undergraduate *automatic control* courses at the University of Sienna since 1999 [9].

Some systems have elements of both simulation and real-world. An example is MIT's I-Lab, which allows real-time monitoring of the deformation of a 31m tall flagpole subjected to wind loading [10].

REMOTELY ACCESSIBLE EMBEDDED SYSTEM LEARNING ENVIRONMENT AT THE UTS

After joining a national consortium concerned with the development of embedded systems in 1999, the Faculty of Engineering at the UTS modified the subject *Operating Systems* so that the laboratory work could focus on embedded systems development. Since 2001, students enrolled in *Operating Systems* have been making extensive use of remotely accessible embedded system hardware and software, as well as a Web-based, peer-supported learning environment.

The environment enables remotely located users to control, monitor and program real embedded system components, and provides users with the tools to undertake real embedded system projects [11][12]. It has three functionally distinct parts, namely:

- A web-based peer-supported learning environment;
- Low-latency video monitoring;
- Remotely accessible embedded system hardware and software.

The Web-based peer-supported part of the environment is a *Blackboard*-based system known at the UTS as UTSOnline. This is a well-established teaching and learning tool that is used in many of the University's subjects. Teachers and learners use UTSOnline to carry out tasks such as distributing and accessing course materials and assignments, and conducting online discussions. The remainder of the environment is (at this stage) specific to the subject *Operating Systems*. A more detailed technical description of the environment has been given elsewhere [12]

A common criticism of remote labs is that students do not get to see the smoke (when experiments go wrong). With this environment, if it smokes, students could see the smoke via the cameras (until the cameras melted!) When this video monitoring was first developed, various cameras were tested for suitability and reliability, and before placing the whole set-up in ventilated cabinets, some cameras were, in fact, melted (but as far as is known, no parts of the system have ever smoked).

An advantage of this approach is that laboratory access is not substituted by simulation. The system makes it possible for remotely located students (and staff) to develop embedded system components that have research applicability or commercial value. Students use the new environment to run and debug programs on real development boards. This enables students to complete most of their embedded system projects off campus (or from general purpose computer laboratories). Students use the environment to complete projects where they become familiar with critical aspects of embedded systems development.

MODIFICATIONS TO THE EMBEDDED SYSTEM LEARNING ENVIRONMENT

Most of the recent modifications relate to the *arbitration* software. This is the software that provides a mechanism for fairly allocating development-board connections to incoming requests. The software maintains a queue of students wishing to initiate connections, and enforces time limits on these connections. It stores resources in a PostgreSQL database. It also performs various other functions, such as recording statistics on average usage per session and hourly usage. This software was developed in-house.

Modifications were made as a result of a re-analysis of the system in the light of observations made during the semester, feedback from users and consultations with other staff. The most significant changes resulted from the need to overcome previous limitations. The superseded system required each connected device to have the same configuration (for example, the same peripherals and same amounts of RAM). This meant that if the altered learning objectives necessitated the implementation of a new board or peripheral (for example, a servo controller as opposed to an analog-digital converter), all of the development boards had to be upgraded (at great cost).

Following the successful implementation of the system, a number of staff were interested in adapting it for various other subjects. However, the previous arbitration system only supported development boards based on the Motorola Coldfire architecture. A more generally applicable arbitration system was thus desired.

Addressing these issues required significant modifications. Support was added for generic architectures and heterogeneous configurations. The new arbitration system now only uses a database for queue management functions and storage of architecture/device configuration parameters.

Device handling functions, such as cleaning up after sessions and allocation (and de-allocation) of the boards, are now performed by separate external scripts. These scripts are easily customisable and, as such, allow much-improved flexibility. It is now possible to develop new configurations rapidly by generating a new set of scripts for each different set-up.

Default templates were developed for configuring and connecting to various devices. These templates can be customised and extended for each connected device via an OO inheritance-based mechanism, allowing heterogeneous configurations. For example, some Coldfire boards can now support analogue-digital converters, while others are connected to servo controllers.

Additional modifications to the arbitrator allow students to specify the device, peripherals and configuration they desire.

Other modifications improve the system's user-friendliness. Students using the previous system sometimes became annoyed when they were disconnected after their time limit expired, even when no one was waiting in the queue. The new system provides automatic time extensions when there is no one in the queue. The duration of the extension is programmable (currently set to five minutes). In addition, students wanted a warning message before their time limit expired. The new system provides a warning message sent to their console at a programmable duration before expiry. This is currently set to 5 minutes.

In the event of a device failure, students using the superseded system would report the problem via UTSOnline, and support staff would generally take the problem-device offline until repairs were completed. However, during these periods, the arbitration system used to continue to allocate students to the failed device. A new feature was added to the arbitration system to allow students to ask the arbitration system not to allocate a specific device.

ASSURING EDUCATIONAL QUALITY

A recent DEETYA report indicates the need for an evidence-based approach to the usage of computer and information technology (CIT) in education. It suggests focusing on … *determining how CIT can cost-effectively add value to students' learning as part of a broader, more flexible and responsive approach to learning design and delivery*, and that *decisions about use should be informed by evidence that, in this particular context, for this particular group* the CIT adds value to learning in a way that contributes towards achieving the objectives of the broader learning programme [13].

Similarly, Patil and Pudlowski put forward the view that comprehensive research of the effectiveness of any Web-based learning environment is required [14]. They suggest various characteristics considered essential for appropriate Web-based engineering education, including:

- User friendliness;
- Appropriate encouragement for self motivation;
- Simple Web delivery methods;
- Learner centred focus;
- Opportunities for learners to test theories;
- Facilitation of active learning;
- Facilities for self-assessment throughout the learning process.

Distinctions between different implementations of remote laboratories are readily drawn in terms of physical or technical differences, for example, systems that connect users to real-world objects, as opposed to those that do not. It is felt it would also be beneficial to attempt to distinguish between remote laboratories' educational aspects. In looking at the UTS' own remote laboratory, questions were asked to determine if the environment adds value to learning. Such questions include the following:

- In using this environment, do learners need to organise their time effectively? Can learners decide what they want to work on and when?
- Do learners have the ability to test theories and different approaches to framing and solving problems? Are learners able to test ideas experimentally?
- Are learners able to assess their progress continually? Can they learn from errors incrementally, gain feedback from and the results of their work instantaneously, and gain insight into their strengths, limitations and areas for further development?
- Is self-motivation is encouraged? Are learners allowed to continue their session if no one is waiting on the queue?
- Do learners work together and communicate with others via the online discussion board resources and e-mail?
- Do learners gain experience in effective communication (especially in using modes of communication other than face-to-face), building networks and developing trust in colleagues?

CONCLUDING REMARKS

It needs to be considered whether there is sufficient data to answer some of these questions. And, these are just some of the *yes* or *no* questions. Ideally, the authors would like to investigate the relationships between design variables and educational-outcome variables with greater rigour. But doing this, of course, would require significant resources in terms of staff-time, or preferably, funding for independent research. So although there is a reliable and usable system up and running, there is still much work that needs to be done in order to understand better the educational implications.

REFERENCES

1. Scott, G. and Yates, W., Using successful graduates to improve undergraduate education. *European J. of Engng. Educ.*, 27, **4**, 363-378 (2002).
2. The Institution of Engineers, Australia (IEAus), *National Generic Competency Standards for Stage 2* (2nd edn). Canberra: IEAus (1999).
3. Gomes, V.G., Choy, B., Barton, G.W. and Romagnoli, J.A., Web-based courseware in teaching laboratory-based courses. *Global J. of Engng. Educ.*, 4, **1**, 65-71 (2000).
4. Edwards, S., Can quality graduate software engineering courses really be delivered asynchronously on-line? *Proc. Inter. Conf. on Software Engng.*, Limerick, Ireland, 676-679 (2000).
5. Lilja, D.J., Teaching computer systems performance analysis. *IEEE Trans. on Educ.*, 44, **1**, 35-40 (2001).
6. Albu, M.M., Holbert, K.E., Heydt, G.T., Grigorescu, S.D. and Trusca, V., Embedding remote experimentation in power engineering education. *IEEE Trans. on Power Systems*, 19, **1**, 139-143 (2004).
7. Sanchez, J., Morilla, F., Dormido, S., Aranda, J. and Ruiperez, P., Virtual and remote control labs using Java: a qualitative approach. *IEEE Control Systems Magazine*, 22, **2**, 8-20 (2002).
8. Ko, C.C., Chen, B.M., Chen, S.H., Ramakrishnan, V., Chen, R., Hu, S.Y. and Zhuang, Y., A large scale Web-based virtual oscilloscope laboratory experiment. *IEEE Engng. Science and Educ. J.*, 9, **2**, 69-76 (2000).
9. Casini, M., Prattichizzo, D. and Vicino, A., The automatic control telelab: a user-friendly interface for distance learning. *IEEE Trans. on Educ.*, 46, **2**, 252-257 (2003).
10. Amaratunga, K. and Sudarshan, R., A virtual laboratory for real-time monitoring of civil engineering infrastructure. *Proc. Inter. Conf. on Engng. Educ.*, Manchester, England, UK, 1-6 (2002).

11. Moulton, B.D., Lasky, V.L. and Carmody, N.J., Practice-based engineering education: a distributed environment for teaching and learning embedded systems *Proc. 5th UICEE Annual Conf. on Engng. Educ.*, Chennai, India, 33-35 (2002).
12. Moulton, B.D., Lasky, V.L. and Murray, S.J., The development of an environment for remote embedded systems: feedback from students and subsequent implementation changes. *World Trans. on Engng. and Technology Educ.*, 2, **1**, 65-68 (2003).
13. Barraket, J., Payne, A., Scott, G. and Cameron, L., Equity and the Use of CIT in Higher Education. Canberra: DETYA Evaluations and Investigations Program (2001), http://www.detya.gov.au/highered/eippubs/eip007/execsum.htm
14. Patil, A.S. and Pudlowski, Z.J., Instructional design strategies for interactive Web-based tutorials and laboratory procedures in engineering education. *World Trans. on Engng. and Technology Educ*., 2, **1**, 107-110 (2003).