

1-1-2008

## **E-Learning Methods as a Driver for the Internationalisation of Web Development Courses**

Justin Brown  
*Edith Cowan University*

Follow this and additional works at: <https://ro.ecu.edu.au/ceducom>



Part of the [Educational Methods Commons](#)

---

### **Recommended Citation**

Brown, J. (2008). E-Learning Methods as a Driver for the Internationalisation of Web Development Courses. Retrieved from <https://ro.ecu.edu.au/ceducom/5>

EDU-COM 2008 International Conference. Sustainability in Higher Education: Directions for Change, Edith Cowan University, Perth Western Australia, 19-21 November 2008.  
This Conference Proceeding is posted at Research Online.  
<https://ro.ecu.edu.au/ceducom/5>

**Brown, J, Edith Cowan University, Australia**  
**E-Learning Methods as a Driver for the Internationalisation of Web Development Courses**

Justin Brown

School of Computer and Information Science  
Edith Cowan University, Australia  
E-mail: j.brown@ecu.edu.au

**ABSTRACT**

Over the past decade, as software development has moved from a platform specific, desktop based software approach to a web focused environment, the ability to develop courses for wider delivery has increased. At the same time university courses have undergone major changes in teaching mode, with an expectation that online versions of teaching materials should be delivered as readily and to the same quality as in-class materials.

This paper examines how the re-development of course materials to support online (off-campus) students as well as on campus students provides an opportunity to deliver those same materials to overseas partners. This is brought about by the need to select programming environments that are readily available to online students, along with detailed learning materials that allow online students to work at the same level of detail as on campus students. The integration of freely available and easily configurable development environments and teaching/assessment items based on those environments provide the opportunity for international teaching partners to support both their staff and students. This support comes from not having to rely on expensive, difficult to configure software systems, and by providing learning materials that are written specifically for those environments, so that both configuration and use of the environments form core elements of the teaching process. Examples of this process from a number of web programming are discussed in the paper, as are the results from both the local and international perspective.

**Keywords:** E-Learning, Web Development, Programming, Teaching Practice

**INTRODUCTION**

E-learning has evolved from a novel method of content delivery used by a small number of web literate staff to a standard mechanism by which universities and training organisations provide access to their learning materials and institutional resources. E-learning has changed and is continuing to change the landscape of higher education and training, especially within the halls of modern universities (Dreher, Cummings et al. 2006). There appears to be an expectation from students, both local and external to any given organisation, that they will be able to study a part or all of a course via the web, entirely removed from the campus and the classroom. This has certainly put the onus on universities and on individual lecturers to ensure that the materials being delivered via the web are done so at the same level of quality as they would be for traditional in-class teaching (Horton 2002; Nelson, Bhagyavati et al. 2005).

With this change in university teaching and learning in mind, this paper will examine the case of a school of computing and information science located within Western Australia, focusing on how the evolution of teaching practice and materials to suit both online (off-campus) and on-campus students has resulted in far more efficient delivery of the school's courses not only to these students, but to the school's international partners.

## **BACKGROUND**

The school in question offers a number of courses, ranging from computer science to library and information science, with qualifications ranging from Diplomas through to PhD. Each course is comprised of a number of units, each unit focusing on a specific topic and running for 12-13 teaching weeks. Each unit usually contains a lecture and a lab / workshop component, with assessments being multiple assignments and an end of semester examination.

From 2002 onwards the school has implemented a policy of developing a nearly ubiquitous e-learning presence of a majority of the units and courses, so that all but a handful of units are now available via the web, delivered via the Blackboard Learning Management System. As well as all lecture and lab / workshop materials being available via each unit presence in Blackboard, audio/video tools are also used to enhance the fidelity of the learning materials. These tools include Camtasia 5.0 for integrated lecture slides/audio/animation as well as screen captured images and video. The units and their resources are regularly audited to ensure that the content, the assessments and all related unit resources are being delivered in such a way as to provide as much as possible a similar learning experience to the online students as to the on-campus cohort.

As well as delivering these courses and units to on-campus (or in-class as they will be referred to) and online students, the school has a number of international partners who purchase the materials, assessments and staff moderation so that the courses can be delivered in their own country to their own local students. These partners are spread from the southeast of Asia through to India and Africa, with each having their own local teaching methods and I.T. infrastructure. This last point was, for some time, a crucial issue when delivering computer science topics internationally, where specialist software installations were required in order to correctly establish a learning environment. Computer labs setup for programming students would often require expensive licenses, specialist server installations and any number of custom settings in order to create an environment in which students could begin their lab / workshop and assessment work. Outside of the campus labs, for both online students and the international partners, creating such a specific environment could be difficult, if not impossible. As the remainder of this paper will show, solving the problem of I.T. environment setup for local online students actually solved the problem for the international partners, at least in terms of the web programming units taught by this author.

## **WEB PROGRAMMING ENVIRONMENT**

During the last decade this author has taught a large number of different units, from basic research skills through to database security and most things in-between. However it is web programming and web application development that is the main teaching focus for the author, and the area in which the most amount of change has occurred in unit content delivery. The three primary web programming units taught by this author have been discussed in previous papers, but in summary they examine client-side markup systems (xml, xslt, css, xhtml), client and server-side web applications (xhtml, php, mysql) and server-side applications specifically (php, asp, asp.net). To simplify the following discussion these three units will be referred to respectively as Markup, Php and Dot Net. This paper does not claim that in the context of web programming these three sets of technologies and environments are the only tools that are available (Barbour 2004), only that they have evolved to meet this author's teaching needs and the needs of his students.

Each of these units has specific requirements in respect to the software tools required for students to actually develop applications relevant to their lab materials and assessment specifications. The following discussion of each unit will detail the development of the unit, the software tools required and the current state of the unit content and technical environment. It should also be noted that whilst in the past the international partners received cd-rom images of the course materials contained within the schools previous in-house learning management system, they now have access to their 'own' version of a unit in Blackboard, copied directly from the last semester's materials as delivered to this

author's local students. This means that lecturers such as this author do not need to keep two systems updated, and that a majority of materials, resources and approaches used for local students can be used for delivery of courses to international partners.

## **MARKUP**

Of the three units discussed in this paper Markup has always been perhaps the easiest to teach online and internationally due to the simple teaching environment required, namely a capable text/html editor and a selection of modern web browsers, such as Internet Explorer and Mozilla Firefox. As there is no requirement for a web or database server, all the code can be written in the editor and parsed via a web browser to display the resulting application. Though over the last five years Markup has evolved from a primarily html oriented unit to a much more xml and xslt focussed offering, it has always been the easiest to deliver outside of the classroom, and in many ways provided an impetus to change Php and Dot Net so as to achieve the same results. Markup can be delivered on any operating system environment that has provision for a graphical user interface and the aforementioned text editor and web browser, meaning that online students and international partners using Microsoft, MacOS or Linux environments can utilise the learning materials to the same level. The lecture and lab / workshop files are delivered as .PDF documents so as to ensure cross-platform compatibility and, where possible, most if not all audio and video materials are delivered in MP3 or Adobe Flash format.

Other course and unit authors looking to design a web development unit for online delivery could certainly do worse than to start with a Markup like unit, where levels of technical difficulty could be staggered from simple html web pages through to advanced xml and xslt driven client side web applications such as in the case of Markup. The limiting factor in the latter case is no longer the technical learning environment, but the ability of the online student or the international partner to support any queries or misunderstandings about the learning materials. This problem is mitigated to a certain degree by the 'extra' materials that this author adds to each unit, such as an audio/video summary of each weeks lecture and lab, so that the online students are alerted to any tips, tricks or queries raised by the in-class cohort. As these same materials will be used for the overseas partners in the semester that follows, these 'extra' files help their students and lecturers just as they help the local in-class and online students.

## **PHP**

In its earliest days the Php unit was extremely challenging to teach on campus, and was not even considered for online delivery (Brown 2006). Initially the unit required a central server on which Perl rather than Php scripts could be run, and in order for students to run their scripts they had to write them on a local machine in the computer lab and then upload them to the central server using ftp. Each student in the class required an account on the server machine and whenever the server went down it had to be manually rebooted in short order to stave off emails from panicked students. Eventually the environment moved to php and mysql hosted on machines in the labs, though the setup was still quite specialised as it required the php plug-in to Microsoft's Internet Information Services (IIS) web server running under XP Pro. This presented further difficulties as students typically had Windows XP Home which did not support IIS, or if they did have XP Pro they did not have the original discs to install IIS from. Whilst Php is a cross platform scripting language which can be plugged into almost any web server on any operating system, setting up Php with the most popular web server in the world, Apache, was still a non-trivial exercise. Even before the unit went 'online' students were trying to set up home web servers so that they could work on their labs and assignments outside of the classroom. Aside from those students lucky enough to have the same systems at home as on-campus in the labs, most would try to cobble together a working php and mysql alternative environment and hope the code worked in-class as it did at home. Such configuration issues caused massive headaches in consistency, from the teaching and learning perspective and from the assessment grading perspective. Mysql also provided numerous problems for this author and students working outside of class as easy to use and install interfaces to the database were not widely available at the

time, whilst the added problem of configuring Php to work properly with Mysql also arose as the two environments gradually diverged from working as closely with each other.

Delivering this unit to the international partners was always extremely challenging and frustrating, both from the perspective of this author and the staff and students overseas who were trying to come to grips with not just the coding content of the unit, but the fundamental setup required just to get started. Each semester required dozens of emails to be exchanged between this author and the local lecturer in order to 'help' get their local systems up and running, systems that were usually very different from those used in the local labs. It was obvious that a solution was required that would work just as well for the local students as well as those overseas, preferably in a way that each student could download a pre-built environment provided to them and just start working. This is exactly the solution that occurred at almost the same time that it was decided to deliver the Php unit online as well as in-class.

At the time a number of integrated web and database development environments began to appear, no doubt to address the issue for novice developers struggling to configure their development environments before even getting to work on code. Of the tools available at the time XAMPP from the ApacheFriends organisation (<http://www.apachefriends.de>) was perhaps the most suitable to all facets of the Php unit. It included a single file installer that set up Php (in later versions both Php 4.x and 5.x), the Mysql database, a web based interface onto the database called PhpMyadmin plus an integrated console to individually start and stop these services (see Figure 1).



Figure 1: Xampp Server and Web Interface

As well as offering an easy to install, integrated system, XAMPP was also cross platform, allowing students to deploy the same system to Microsoft, MacOS or Linux operating systems. At the time that XAMPP was adopted as the primary scripting/web server/database server environment for the Php unit, the unit itself was totally re-written with XAMPP as the basis of all lecture examples and all lab materials. All assessments were to be marked using XAMPP, giving students confidence that if their assessment code worked for them, it would also work for their instructor. The very first lab session in the unit was spent getting students to download XAMPP and install it, locate all the required folders for the web and database servers, and explore the PhpMyadmin interface. As well as being provided a link to the XAMPP website where they could download the latest configuration of XAMPP, a pre-configured version was put together by this author, specifically designed to run on student home machines as well as in the university labs.

The transformation in teaching the Php unit, both to the local online students and the international students was quite remarkable, with the issue of configuration disappearing almost immediately. In-class students could work at home, online students could work at home and the international students could run whichever version of the server they liked on either their local campus computers or home computers, all with the same consistent application architecture. The level of work being returned from all students, regardless of delivery mode, has improved significantly with the re-write of the unit, where the emphasis can now be purely on the coding and not on spending weeks trying to configure a workable development environment. Perhaps the most telling result of this move to online teaching and the sourcing of such an effective development environment is that not a single email exchange has

been required between this author and any overseas partners in relation to setting up the technical environment for the Php unit. With the addition of the 'extra' materials as discussed previously, alongside an exceptionally good textbook, the transition to online delivery has been extremely successful.

## **DOT NET**

This unit was given to this author to teach relatively recently and at the time required a total re-write, a process that benefited greatly from the experiences in the Php unit. As the Dot Net unit was designed to give students hands on skills in a number of different server-side environments, so that they could compare and contrast tools and methods, the ones chosen were those that offered the ability to be deliverable online and internationally. For the first half of the unit students are exposed to Php and Asp, and are required to develop and deliver a specified application in both environments. Obviously the Php component was no problem as the server environment and teaching materials from the Php unit were easily integrated into the unit's resource list. The Asp aspect of the unit was more of a challenge for, as with the earlier versions of the Php unit, many students did not have the necessary operating system to run IIS. This problem was largely overcome by pointing students towards the BabyWeb server, which is a tiny, free web server that runs nearly all of the necessary Asp commands and is a single file install. Though it is not cross platform, it still allows any student running any version of Windows to develop their Asp labs and assessments.

For the second half of the unit students are required to develop and deliver an assessment using Microsoft's Asp.Net development environment, in this case assembled using Visual Studio 2005. While Visual Studio is licensed software and too expensive for most students to purchase for one assessment, Microsoft has joined the trend of providing 'lite' versions of big software, this time in the form of Visual Web Developer 2005 (and now 2008). These environments have the Asp.Net tools only rather than the entire range of Microsoft languages and platforms. Included with this free version of the Asp.Net environment is the Microsoft Development Environment (MSDE), essentially a 'lite' version of Microsoft's Enterprise database system Sql Server.

Students are provided links to the 'lite' versions so that they can download them, whilst all lab materials are written using Visual Studio 2005, but using only those features that are available across all versions. To date, local online students and international students have had no issues in downloading and configuring the Asp.Net environment, and whilst the Dot Net unit is not as cross-platform as the other two development units, this has as yet not proved to be an issue.

## **ONLINE THINKING**

This author has argued in other papers that by targeting teaching and learning materials and associated resources specifically at online delivery, with the aim to create a learning experience that as closely as possible parallels that found in-class, all possible student cohorts will benefit. The in-class students are probably over catered for as they have the dual benefit of on the spot instructor assistance as well as access to the 'extra' resources aimed at bridging the cognition gap for the online students away from class (Garrison and Anderson 2003). The online students have access to the same teaching and learning materials and technical environment as to the in-class students, plus the 'extras', driven by the experiences of the in-class students. The international students and their instructors enjoy the same benefits as the online students, typically in the semester following that cohort. The best practice elements developed for the local online students, where feedback regarding materials and delivery can come in on a weekly basis, is rolled into the international materials in a constant plan->do->review cycle.

Of course the obvious issue that arises in this particular instance is that such an iterative cycle of improvement requires a great deal of effort on behalf of this author and presumably other academics

committed to providing as high a quality learning experience as possible to those outside the classroom. Computer science topics, particularly web development, are heavily disposed to constant change and rarely provide the opportunity for static content (Venkat, Jagadeesh et al. 2006), assessment or teaching practice.

By ‘thinking online’ academics can essentially kill three birds with one stone, catering to their in-class, online and international partners, rather than trying to treat each as a separate entity. Some authors in the literature have gone further, customising course materials on a per-country basis (Lee 2002), which in the context of the units discussed here would be untenable. It is this author’s experience (and belief) that treating course materials, assessment and teaching practice differently for the three cohorts identified in this paper dilutes each cohort’s learning outcomes and actually creates more work than the practices identified above. Figure 2 below represents elements of this online focus along with the plan->do->review cycle, where capturing events from in-class teaching informs the ‘extra’ materials to close the fidelity gap for the online students, which in turn feeds into a more comprehensive, self-contained curriculum for the international partners.

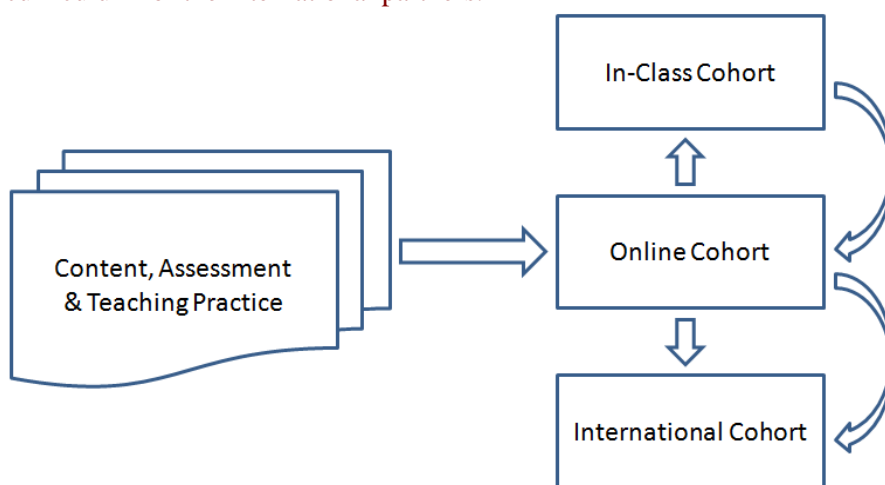


Figure 2: Online Content, Assessment and Teaching Practice Model

This concept of self contained curriculum is particularly important when looking to deliver units and courses of teaching to international markets, as by and large it would seem unreasonable to say to a partner “here is all the content, but as far as teaching practice and environment is concerned, you are on your own”. By designing units to contain high quality content, with established development environments included, teaching delivery can then be largely driven by the students themselves. In such a model, the academic staff member assumes their appropriate role, one of a learning facilitator, not instructor of rote learned training. It is this authors experience that the greater the input into online course development, the greater the learning outcomes for any students who consume those materials.

## CONCLUSION

Most lecturers in most universities in most countries around the world are no doubt well along the path to transitioning their own particular in-class learning materials and teaching practice into the online medium. This paper has examined the case of one such lecturer who has spent half a decade or more trying to develop a comprehensive framework of content, assessment, teaching practice and supporting resources that not only satisfy the more challenging learning scenario experienced by online students, but by all students. In this process of addressing online teaching and learning the often vexed issue of delivering highly technical, computer science based curriculum to international partners has, to a significant degree, resolved itself.

International partners utilising this author’s unit materials receive the same materials, same assessments and same technical environments as the local in-class and online students, and apart from some differences in procedures for submission and marking of assessments (which is conducted at the partners end) these units now run as easily overseas as they do locally. This makes the delivery of such content to international partners much easier for this author, for the school, for the partner’s lecturing staff, and above all, for the international students themselves. While this paper does not claim to any radical reform in online teaching and learning, what it attempts to demonstrate is that a focus on completeness and consistency of learning materials, assessments and resources for online units of study will generate tangible benefits beyond just the online audience (Dreher, Cummings et al. 2006). Table 1 below outlines some of the most basic constituents of technical units (i.e. web programming) that this author adheres to when developing content for online and international partners.

<b>Lectures</b>	<ul style="list-style-type: none"> <li>• Where using screenshots, should use the ‘environment(s)’ used in the computer labs</li> <li>• Documents as .PDF documents</li> <li>• Do not use colloquialisms that international students will not understand</li> <li>• Provide audio ‘summaries’ alongside the lecture files as individual .MP3 files (and refer to each slide number when discussing the slides)</li> </ul>
<b>Labs / Workshops</b>	<ul style="list-style-type: none"> <li>• Spend first week getting students to set up, install and test their working environments</li> <li>• All lab / workshop / tutorial sheets should use the specified development environment for all examples and screenshots</li> <li>• If at the end of any in-class lab / workshop something particularly difficult is encountered, discuss it in an audio/video ‘extra’ and put it up for the online students promptly (within the same week)</li> <li>• Each week should be scaffolded, so that a single, consistent ‘system’ is built over the period of several contiguous labs / workshops, rather than totally unrelated topics each week (Paxton and Heredia 2004; Athauda, Kodagoda et al. 2005)</li> <li>• Do not have separate lab / workshop materials for in-class and online students</li> </ul>
<b>Assessments</b>	<ul style="list-style-type: none"> <li>• Assessments should be relevant to the concepts and skills covered in the labs / workshops. As an example, if an assessment requires an application to add, edit, update and delete data to a database from a web form, those skills need to be covered before that assessment is due</li> <li>• Assessments are to be marked in the specified development environment as provided to the students</li> <li>• The assessments should have requirements that match the lab / workshop materials, but not be based on the same examples (i.e. making students apply the same concept in a different context)</li> <li>• Provide for ‘extra marks’ in any assessment so that skills and concepts above and beyond those taught in the lab / workshop can be added by particularly capable students</li> <li>• Do not have separate assessments or assessment requirements for in-class and online students</li> <li>• Assessments should be authentic and not off a type that could easily be downloaded from free ‘scripts’ sites (Brown 2006; Wang 2006; Brown 2008)</li> </ul>
<b>Resources</b>	<ul style="list-style-type: none"> <li>• Provide the development environment, or links to the environment within the online materials</li> <li>• Ensure that development environment is free, and where possible, cross-platform</li> <li>• A textbook with examples that take students through building working applications is essential for online web programming units – though can be hard to find (Sahama, Ho-Stuart et al. 2004)</li> <li>• Make heavy use of online tutorials from the myriad of web coding sites</li> <li>• Create ‘tasks’ which students must complete after reading these tutorials “associating an activity with a curriculum”, p. 1101 (Kam, Ramachandran et al. 2007)</li> </ul>



*Table 1: Minimal requirements for units developed for online and international students*

Those facing the challenge of creating an online presence for their technical, programming related courses of study will hopefully find some of the previous discussion useful, and perhaps even applicable to their particular situation. The initial process can be extremely time consuming, especially in terms of materials writing and selection / integration of the requirement technical environment. Whilst this paper has really only scratched the surface of what has been the ongoing, iterative work of upwards of a decade of teaching web programming, hopefully it does show that the investment in the time and effort required to achieve high quality online learning is indeed worthwhile. This is particularly true when considering that a correctly designed curriculum and its resources will address the needs of both a 'local' online community of students and the greater community of students found in most international partnership programs.

## REFERENCES

Athauda, r., N. Kodagoda, et al. (2005). Integrating industrial technologies, tools and practices to the IT curriculum: an innovative course with .NET and java platforms. Proceedings of the 6th conference on Information technology education. Newark, NJ, USA, ACM.

Barbour, R. (2004). "Postmodern critique of teaching and learning support systems at an internet distance." SIGCAS Comput. Soc. 34(2): 2.

Brown, J. (2006). Teaching Web Applications Development in a Fully Online Environment: Challenges, Approaches and Implementation. 2006 International Conference on E-Learning, E-Business, Enterprise Information Systems, E-Government, & Outsourcing, Las Vegas, USA, CSREA Press.

Brown, J. (2008). Assessment Practice for Mixed Mode Delivery of Web Programming Topics: increasing learning and reducing cheating. 2007 International Conference on Frontiers in Educations: Computer Science & Computer Engineering, Las Vegas, CSREA Press

Dreher, F., M. Cummings, et al. (2006). "The role of IT policies in the CS/IS curricula." J. Comput. Small Coll. 22(2): 267-273.

Garrison, D. and T. Anderson (2003). E-learning in the 21st century : a framework for research and practice. London ; New York, RoutledgeFalmer.

Horton, W. (2002). Using E-Learning. Virginia, American Society for Training and Development.

Kam, M., D. Ramachandran, et al. (2007). Localized iterative design for language learning in underdeveloped regions: the PACE framework. Proceedings of the SIGCHI conference on Human factors in computing systems. San Jose, California, USA, ACM.

Lee, J. (2002). Internationalization of the curriculum report of a project within computer science. Proceedings of the 33rd SIGCSE technical symposium on Computer science education. Cincinnati, Kentucky, ACM.

Nelson, M., Bhagyavati, et al. (2005). "Online teaching practices (both best and worst)." J. Comput. Small Coll. 21(2): 223-230.

Paxton, j. and k. Heredia (2004). Providing curricular assistance to a third world university computer science department. Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education. Leeds, United Kingdom, ACM.

Sahama, T., C. Ho-Stuart, et al. (2004). Developing and delivering a software internationalisation subject. Proceedings of the second workshop on Australasian information security, Data Mining and Web Intelligence, and Software Internationalisation - Volume 32. Dunedin, New Zealand, Australian Computer Society, Inc.

Venkat, N. G., N. Jagadeesh, et al. (2006). "A learning-centered approach to designing computer science courses." J. Comput. Small Coll. 21(4): 96-103.

Wang, A. (2006). IT education in the flattening world. Proceedings of the 7th conference on Information technology education. Minneapolis, Minnesota, USA, ACM.

## **COPYRIGHT**

[Justin Brown] ©2008. The author/s assign Edith Cowan University a non-exclusive license to use this document for personal use provided that the article is used in full and this copyright statement is reproduced. Such documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. The authors also grant a non-exclusive license to ECU to publish this document in full in the Conference Proceedings. Any other usage is prohibited without the express permission of the authors.