The Uptake of CAA: Some Theory! Some Practice!

Steve Bennett, Learning Technology Development Manager, LTDU, University of Hertfordshire
Andrew Oliver, Multimedia Consultant, LTDU, University of Hertfordshire
Ashley Pinn, Lecturer in the Faculty of Art and Design, University of Hertfordshire

Abstract

The uptake of CAA is primarily discussed in institutional terms, the establishment of software standards, invigilation procedures, the solicitation of questions and their warehousing in *question-banks*, liaison between academic staff, technicians and the exams office – in short a systemic totality of small initiatives, the result of an unrelenting bureaucratic heroism.

However there is also another side to all this: what gets the instructor into technology in the first place? What inspires? And what keeps him inspired, both in the technology and their own subject? We don't, after all, want to give rise to a corps of gadget-freaks and media wannabes, but rather great teachers who find new ways to expound, disseminate, problematize and assess their subject through technology.

We might call this a *process of uptake* and we propose to examine it at its most formative moments, where the curiosity of academics is concretized into real multimedia production. We will look at the academic staff/learning technologist nexus and examine how this can facilitate such an uptake. Finally we will put forward a series of considerations at the level of assessment software which ought to be borne in my before commencing such an enterprise.

The *process of uptake* we document here occurred through a project undertaken by the University of Hertfordshire's *Learning Technology Development Unit* and a lecturer from the University's School of Art and Design, Ashley Pinn. The project began after Ashley had obtained money from the University's *Multimedia Development Fund*, and he began to work with a multimedia consultant from the *LTDU,* Andrew Oliver. In the process of doing the project, creating web pages followed by large amounts of quizzes, we began to discover that none of the available solutions really suited our purposes, as far as developing the quizzes were concerned. Therefore another member of LTDU, Steve Bennett, then became involved in writing a question generation shell, which was then in turn used by Andrew, and finally by Ashley himself as well as in numerous other projects. In order that the specificity of each contribution be registered, we have decided to present this paper in the form of three points of view.

Academic Instructor: Ashley Pinn

Workshop Techniques is an on line tutorial resource designed to teach the traditional subject of wood and metal working to undergraduates at the Faculty of Art and Design, University of Hertfordshire, (http://www.herts.ac.uk/ltdu/projects/mm2). The application makes use of embedded diagnostic testing to assess the student's knowledge of health and safety issues and test for skills competency in using the workshop machines. It was built to harness the students' enthusiasm for the Web and Email in order to teach what might be considered a traditional subject (Woodworking and Metalworking) in a new hi-tech way.

Workshop Techniques was developed to reduce staff workload but without compromising crucial formative support towards students. Rather the aim was to create a resource which is readily available to support the students in their learning - particularly outside normal staff-to-student contact hours. Over the last few years staff from the Faculty of Art and Design have spent an increasing amount of their time providing one-to-one tuition to students who were absent from crucial workshops. However due to the large number of students attending the course it is impossible for staff to provide such an intimate level of support on an individual basis. In addition the staff member is often unsure as to what knowledge or skills the student is lacking in and finds it difficult to structure the remedial session.

In the development effort, the first thing was to establish methods of communicating efficiently between myself and Andy (Andrew Oliver the multimedia consultant). At this stage I was on a distant site some 2 miles away from his department. The first thing to master was the use of email attachments for the transfer of the package's text. This text was generated first by WordPerfect 5.0 (this was all I had available) and then by either ClarisWorks or MS Word. Andy then transformed it into web pages.

The next step was transferring Photographs of the workshop machinery to Andy for inclusion in the package. I had worked out that I required at least 350 separate images. To start with I took some images with a 35mm camera I would then have developed, scan and send them off to Andy by FTP. This was incredibly long winded. Instead of instant feedback of whether an image was good enough I had to wait for the images to come back from the developer. Therefore the next stage in my uptake of the technology was learning to use a Digital Camera.

There was a steep learning curve at the beginning, but it didn't last long. Once taken, images from this source could be fed to Andy on a same day basis with very straightforward pre-checking by me. This meant that if an error was found it was easy and quick to take another shot to put it right.

The process of communicating via email, ftp and occasional meetings proved very successful. We were both in a position of knowing what we wanted to do and were searching for the most suitable technology with which to do it. Both of us had to learn new things but could offer each other mutual support in the

event of difficulty. I would encourage any other academics to follow the same path as I did in attempting to use new technology in this way. I do not think the collaboration would have worked so well if I had just given Andy the material and told him to get on with it.

Describing the package itself, there are 10 test sections within Workshop Techniques each designed to test the student's knowledge for a specific machine. Each test section is composed of four question types: multiple choice question, true/false statement, a "fill the gap" exercise where students are required to type in missing terms and machine labelling quizzes. In total the application contains 185 questions. The process of producing a typical test section first involved me conceiving the questions along with their associated distracters, storing them within a Word document and ftping the file to the Andy . He would then copy the questions from the Word document and produce the interactive quiz for the web. This quiz was written in such a way that the results of anyone who took it would be emailed to me.

Because use of the package was an in-built part of a course take up by the students was very high. Some 90% of the student used the package, particularly the tests. Use of the tests resulted in the generation of over 1,000 e-mail responses. What was unexpected was the amount of time each individual student spent on using the package. Figures of 7 to 8 hours were not uncommon. The number of hits the pages got at different times of day was also interesting. The largest number of hits was recorded between midnight and 1am and Sunday afternoons between about 2.30pm and 4 O'clock . This was surprising as prior to the package being developed students only attended a 1.5 hour lecture/demonstration and did a simple multiple choice test based on a slide show.

One of the most encouraging things when developing on-line applications is this kind of tangible feedback one can get about one's teaching and materials. When you see all the email coming in, detailing one's students' understanding of the materials, as well as more interesting findings of the kind described above, it rewards all the effort you put in. When thinking about encouraging the uptake of computer assessment, demonstrating such tangible benefits is an obvious consideration.

Such instant documentation had many uses: you could immediately see whether a student was competent to be able to use the machines on their own. This was valuable for the faculty because Health & Safety requirements demanded that each student was deemed to be competent to use the machinery. Instead of long winded tests in a lecture setting (which had to be physically marked) the package instantly told me the percentage each student got right on each straight-away, so there was no lead time between the student taking the test and being documented as having passed (or failed) the test. This feature was also useful if new students were enquiring about joining or applying to the course they could be given the web address and take the tests before they got here. You could then assess what they have been taught/not taught prior to coming for interview. Indeed at some future stage it could be used as part of the interview procedure in itself.

On a personal level during the doing of the project, I learnt a lot about communicating efficiently via the web (email and FTP). Also the process of writing multiple choice quizzes, involving the generation of distracters for multiple choice tests. My personal profile has been heightened in my Faculty and elsewhere. I have attended a teaching and learning conference in Jacksonville Florida and appear in an article in the LTDU newsletter. I also presented the package at a one day conference on IT at the University of Hertfordshire.

Multimedia Consultant: Andrew Oliver

Collaboration between Ashley and myself began in April 1998. The initial phase of the project involved the production of the theoretical content of the application and continued for 3 months. An integral part of the design process involved the re-evaluation of the current course content and how it was taught. It became evident that in order to be an effective formative part of the curriculum the application would have to have a high image content.

The learning content covered the usage of ten machines and, due to that and the high density of images, a frameset structure was adopted. This allowed me to conceive a structure whereby the navigation buttons (e.g. options to choose machine type, test section etc.) would be permanently on display to the user and be accessed at any point throughout the material - even during the test section. Thus the nature of the project altered from one in which the test sections were to be placed and accessed at specific points within the material to one where the questions could be accessed freely.

This frame based approach provided an “open” learning environment and suited the requirements of the project in that the students were able to test their current knowledge of the subject whenever they wished and thus gained some sense of their progress. Further detail on the development of Workshop Techniques is beyond the scope of this paper however the interested reader is referred to Pinn, (1998) and Oliver and Pinn (1998).

At the outset of the project it wasn’t totally clear what software would be used to create the assessment sections. The public domain package *Web Course in a Box* was tested and seemed to be adequate to the task but only seemed capable of multiple choice single answer questions, or fill in the blanks sentences. The *Castle* project was also inspected, but this, like Web Course in a Box, had most of the marking undertaken at the server side. As the demand had arisen for distribution by CD-ROM as well as over the web, specifically for colleagues in the USA, we needed something which would also function on the client side alone if possible.

Moreover, we wanted something whose look and feel would be entirely controllable by a developer in order that its questions could be seamlessly embedded in a teaching package. In the end therefore, LTDU developed its own question authoring system, the *Question Generator* which could produce the complete test in the form of a single html file. Results from students sitting the test were sent to Ashley as emails via a CGI script on the University

of Hertfordshire's web server: however, even if the person using the system had no web connection and used a CD-ROM, the quizzes could still be used for self assessment purposes.

The *Question Generator* works by the author writing the questions on a web form, following a simple syntax which are then posted to a cgi-script which returns an HTML page with the quiz encoded within it in javascript. Once this was done, I would then incorporate the resulting quizzes into a frameset template using Macromedia Dreamweaver. The final test page would be nested as a secondary frameset within a parent frameset, (which displays the navigation buttons). A typical test section took approximately 1 hour to produce.

A general point to be made here is that *flexibility-of-deployment* needs to be a very important consideration when using any kind of computerised assessment. The number of possible contexts in which one may wish to include some form of assessment is huge and therefore any system which is not too exigent in terms of look and feel, and precludes as little as possible in the areas of customisation and modifiability is to be preferred

Each test section produces an overall score and a summary of the results for each question. On the basis of the scores the student can decide whether they are able to move on to study another machine or should re-visit certain sections. The score is used as a means of determining the students direction of learning through the package. In this way Workshop Techniques is responding to the students misconceptions of the topic and redressing them by indicating the areas of the topic which the student is weak in and should re-consider. The use of diagnostic testing for directive study enables a class of students to come to a common learning point within the course material regardless of either their ability or prior knowledge of the subject

After completing the test the student would submit their answer to the program for scoring. At the same time a total score along with the time at which the test was taken is forwarded by email to the tutor. This serves a number of purposes: first the staff member can actually see if the package is being used at all; second they gain an idea which students are actually using the program and who are not and thirdly they can monitor the score for the whole class. If the staff member finds the class to be struggling they can arrange for extra tuition in the form of revision lectures and practicals.

The use of embedded testing within Workshop Techniques lends itself to three types of usage:

- First the embedded test along with the rest of the application forms part of an integrated course where Workshop Techniques is supported by the more traditional lecture and practical. In this context the application acts as an on-line resource for independent study. Workshop Techniques is supported by the traditional methods of instruction and in turn reinforces the concepts introduced within the lecture.

- Second the application is used as a means of determining the gaps within the knowledge of students who have either missed the practical / lecture sessions or who are struggling with the course content. From the student's test scores the staff can see in what areas of the course the student is having problems with and determine whether the student does indeed have to attend a practical and if so the length and extent of the session. In this context the staff is able to determine the degree of traditional support appropriate to the student's needs
- The presence of overseas students within the course has been steadily increasing over the last few years with many individuals seeking placement beyond the first year. Under these circumstances Workshop Techniques is used within an "interview" context whereby the eligibility of the student to enter either the second or third year of the course is ascertained. On the basis of the test scores the staff can decide whether the student can be moved up and if so what type of additional support they may require.

In addition to the above Workshop Techniques is used for student self study and revision. When evaluated by questionnaire 89% of the students stated that they would consider using the program for further study, (refer to Oliver 1999 for more detail on the evaluation of Workshop Techniques). Analysis of the pattern of usage indicates that they students prefer to use the application on Sunday afternoon and between midnight and 1 am during the week, (Oliver, 1998).

Programmer: Steve Bennett

My involvement came about through the realisation that there wasn't much available for the creation of web based quizzes which were not a) marked on the server side and b) obliged a rather "rustic" look and feel to the questions, not altogether appropriate when delivering a course to Art and Design students.

Our solution was the development of the *Question Generator*. This is a web form into which authors put their quizzes, which would then be submitted to a CGI-script which would return a web page with those questions and the marking, solving, feedback and reporting routines invisibly encoded around them. This question generator owed an enormous amount to a number of university projects being run at the time particularly *TML*, *Castle* and *Euromet*.

The initial trigger in this direction was an email sent to the Computer Aided Assessment mailing list by Helene Missou, then working at the University of Bristol as part of the TML project. As an aside to a response for sources on Web Assessment, she wrote:

> *Also, for another project, I've been doing experiments on using JavaScript for developing courseware materials. Feel free to reuse those scripts (I will add a "how to use" section soon...)*
> http://www.ilrt.bris.ac.uk/~ethm/jscripts/js-intro.htm (10.12.97)

What was contained on that page were some wonderful uses of javascript for the purpose of formative assessment, using Herge's Adventures of Tintin as content (though they seem to be inaccessible at the time of writing).

While the documentation was very good, it nonetheless did require a very good knowledge of html forms and javascript to be able to write questions in its format confidently. However, it was also coded very elegantly, seeming to invite its own automation, to prefigure a kind of simplified scripting environment that would generate these javascript routines and procedures by itself, and encode the uncomplicated input content within it.

We based our simplified scripting environment on the Castle project and while we required a client-side marking system (unlike Castle's server-side implementation), we liked the interface to the Castle question creation system. Essentially a multiple choice single answer question was simply created by a line for the question, more lines for the alternative answers, the correct one being preceded by an asterisk. How perfect this seemed! - the look of the question while being written would look almost identical to a paper equivalent. Additions we added were the ability to put feedback after the interrogative sentence (separated by a pipe symbol |) and the marks awarded for that particular question after the feedback. Therefore a typical multiple choice single answer with 5 marks would look something like this:

> What is the most appropriate meaning of *uptake* to be used when talking of the *uptake* of computer aided assessment? |Incorporation into an organism, be it the academic mind, or the institution he or she inhabits, is the most germane to our enquiry|5⌫
> Understanding, as in *slow on the uptake*⌫
> A Flue which carries hot air upwards⌫
> *Incorporation into a living organism⌫

(Symbols indicate hard line breaks as opposed to word-wraps).

So how was this Castle-like scripting interface to generate a client rather than server side marking system? Through an adaptation of Missou's javascript automatically generated by the CGI-program. This was a Perl script which would parse the question content, and then feed it to an HTML pre-processor similar to that used in the *Euromet* project. In layperson's terms, this meant creating a series of macros which would contain the majority of Missou's javascript and form types, into which the content, the questions, the alternative responses, the feedback and the marks awarded would, upon compilation, be silkily interwoven.

To start with we simply used multiple choice question, single or multiple response, of text or of images, as indeed did the Castle project. However, among Missou's pages there were other types. The cloze or fill-in-the-blanks test for instance which was very easy to *castle*.

At this point the list question was requested, which is to say, a group of combo boxes (a.k.a *drop-down menus* or *choice boxes*) asking the user to order the elements. This was how Andy phrased the request to me:

> *The drop-down menu question should adopt a style similar to that of the others. Namely there has to be a Mark answer, Show Solution and Summary Results boxes. The marked answer box will provide instant feedback to the user on which options were wrong and which were right etc.*

At first sight it seemed this might be more difficult to *castle* than the other question types. Lets take a silly example, namely asking the student to order the numbers 1 – 5 in Italian: uno, due, tre, quattro, cinque. This would involve 5 drop-down menus, each with the 5 elements randomly arranged. Eventually we got the script to look like this:

```
Put these words in numerical order|Go to the next question⌫
Uno⌫
Due⌫
Tre⌫
Quattro⌫
Cinque⌫
```

Where the order would be correct in the scripting form, but which the *Question Generator* would jumble and randomise in their successive incarnations in the dropdown menus which the student would have to correct. Here we attempted to use Castle style simplicity of authoring to generate something, the ordering question which is always incredibly tedious to encode.

The above probably seems like a tedious digression but it illustrates an important point relating to the uptake of CAA. Any assessment engine or similar software system needs to be highly responsive to the requirements of those who use it. It needs to be in a state of continual evolution, since it is impossible to foresee all the possible contexts one might wish to assess, and what cognitive capacities and knowledge one seeks to quantify. Other features added to the software as a consequence of user recommendation have been: the incorporation of equation questions using the public domain HotEqn applet & LaTeX syntax, control by authors over the colour and appearance of the page at compilation time to avoid unnecessary post-editing, finally differential scoring for the questions, all of which needed to be added without compromising the fundamental design consideration of the whole system: absolute simplicity of use.

The Question Generator – where next?

The quizzes produced by the Question Generator, first used in *Workshop Techniques,* were also used in a Macromedia Director tutorial, a project covering *Core Skills in the Humanities* (a test on the ordering of references), and a project in Pharmacology. It was also used in classes *on Technology in Learning and Teaching* which is given to academic staff at the University of

Hertfordshire. Essentially the last lesson of the beginner's class, and the first of the intermediates was dedicated to producing a web based quiz using the *question generator*. This was offered to people with only a minimal understanding of HTML and with no experience of Computer Aided Assessment at all in most cases: most of the attendees managed to produce a web based quiz within an hour.

They did have some difficulties however: some wondered how it was possible to save such quizzes – which wasn't too difficult to answer – go <File><Save As> and you're laughing. Some took a while to understand the semantic function of the pipe symbol, but again that was quickly solved. Moreover, some having got their initial quiz out of the way, almost immediately came up with suggestions for the *Question Generator*: differential feedback according to answer selection, putting line breaks within the questions, reporting the answers to databases rather than email, flash animations within questions and maybe even within feedback!

However, this road ahead constitutes an unpassable path for us. Any further options to the questions might render the syntax too difficult for the naïve user to be able to author quizzes. It thereby becomes more specialised, aimed at the adept rather than the neophyte. And if its aimed at the adept, a greater degree of rigour is required in the maintenance and documentation of the system than we have the time to undertake.

Therefore we offer the code for anyone who wants to make use of it and take it further. And yet, for real *uptake of CAA*, by which we mean the *incorporation into the normal practice of the university*, a way has to be found to please both the adept and the neophyte. We think the *Question Generator* is a good solution for latter: as for the former, we're working on it!

References

Question Generator:
http://www.herts.ac.uk/ltdu/development/question_generator
Workshop Techniques: http://www.herts.ac.uk/ltdu/projects/mm2

Oliver A., W., & Pinn, A. J. (1999), Partners in Crime, Techne-Logos Vol 1, LTDU, University of Hertfordshire

Pinn, A. J. (1998) Woodwork Tuition for the Computer Literate Student , paper given at
10th International Conference on College Teaching and Learning Jacksonville Florida

Missou, Helene "Re: software for constructing tests.", online posting. 10.12.97. UK computer-assisted-learning Discussion List

Euromet: http://www.euromet.met.ed.ac.uk/

TML/NetQuest: http://www.ilrt.bris.ac.uk/mru/netquest/tml/

Castle: http://www.le.ac.uk/cc/ltg/castle/

HotEqn:
http://www.esr.ruhr-uni-bochum.de/VCLab/software/HotEqn/HotEqn.html