Information Technology and Legal Education: Towards 2000

9th & 10th April 1992
Sponsored by
Context Ltd; Linklaters & Paines; Needham & James

Computer Assisted Legal Training for Solicitor and Barrister Students in Northern Ireland

Mary McAleese, Patrick Brannigan and George Munroe

The Queen's University of Belfast.

Keywords: Courseware - Criminal Procedure -Design -HyperCard -Project Team
The following acronyms are used frequently within the text:
- IPLS Institute of Professional Legal Studies
- CCBL Centre for Computer Based Learning
- QUB Queen's University Belfast
- CALT Computer Assisted Legal Training
- CPI Criminal Procedure - Indictable
- MC Magistrate's Court
- SWIG Student Workstation Implementation Group

Abstract: We describe the development of two computer based packages to teach Criminal Procedure. The HyperCard system on Apple Macintosh computers was used. The first package has been in use for two years with barrister students and we report related observations. We offer comments, based on experience, about the issues involved in a computer based approach to training.

Outline: Introduction

Background; The Scope for Computers in Legal Training, The Realisation of CALT, A Project Approach, Courseware Subject Matter

Development; Design of CPI, Production of CPI, Description of Completed CPI Package, The MC Package
Implementation; Use of CPI, Evaluation of CPI
Discussion

Introduction

Computer Assisted Legal Training (CALT) has been realised in this University through the

introduction of a computer based teaching package (courseware) into the existing training framework. We have developed this package ourselves and are currently completing a second. The first is entitled 'Criminal Procedure -Indictable' (CPI), and the second 'Magistrate's Court' (MC). In what follows we provide details of the progress of CALT to date. We first consider the background factors which prompted and enabled the initiative. This is followed by a report on the actual courseware development, including the initial design, how it was produced, what the completed CPI package consists of, and how and why the design of MC differs from that of CPI. We then describe the implementation of the courseware, how it is being used and how we are trying to evaluate its effectiveness. Finally we try to bring together a summary of the important issues in a discussion section.

Background

The Scope for Computers In Legal Training

The Institute of Professional Legal Studies enrols 20 Bar students and 70 Solicitor students annually in a postgraduate vocational training programme designed to prepare them for careers as practising lawyers. The course concentrates on preparing students for the early years of practice. It covers the main areas of civil and criminal practice and procedure, everything from drafting a will to defending an alleged rapist. The course has a considerable skills acquisition orientation with dedicated courses in draffing, advocacy, client counselling, negotiation, word processing and Lexis training (Lexis is an extensive computer based national and international law library).

The Solicitor's office or Bar library structure within which most of our students will eventually work, does not currently call for computer literacy. A small number of barristers use word processors; a small number have been trained to use Lexis, though many still access it via a mid-user system. Solicitors' offices are moving rapidly towards the extensive use of computers though almost invariably for the purposes of word processing. Less than 10% of our students have any computer experience. From discussions with students, staff and members of the profession we identify a 'fear barrier' in relation to computers allied to an almost contradictory recognition that they are the tool of the future and must be harnessed to the use and benefit of lawyers.

We wanted to address that fear and to see if we could provide an effective computer based teaching program or series of programs which would introduce students easily to computers and which could complement, supplement or even replace conventional teaching programs.

The Realisation of CALT

The real incentive for the launch of CALT at IPLS in 1989 came from the Student Workstation Implementation Group (SWIG) initiative. This was jointly funded by the Department of Education for Northern Ireland and the Industrial Development Board, and offered the opportunity to departments at Queen's University Belfast (QUB) to develop the use of computers in their teaching programs. SWIG provided the hardware (in our case three Apple Macintosh computers) and CCBL provided the necessary courseware production experience. Apple kit was chosen because of the intrinsic and excellent user interface which enables the most apprehensive of users to gainfully employ the computer without the 'mysterious art' of computing getting in the way (Apple Computer Inc., 1987).

The realistic costing of courseware development was important as was the selling of the idea to both CCBL and the Council of Legal Education (the governing body of the Institute - it is a partnership of the Law Society of Northern Ireland, The Honorable Society of the Inn of Court of Northern Ireland and The Queen's University of Belfast). The support of all parties was enthusiastically given, and an ad hoc working party was set up.

A Project Approach

The production of courseware was regarded as a 'project' from the start. Members of IPLS and CCBL formed a project team and meetings were arranged where the views of all parties could be discussed. Clearly defined objectives were agreed and a set timetable drove the work relentlessly. Regular joint reviews of the product and progress moulded the team consciousness and avoided lawyer and programmer demarcation disputes. A lot of time was spent simply listening to each other.

The CPI project team consisted of the following individuals: Mary McAleese, IPLS; Patrick Brannigan, CCBL; Barry Valentine, IPLS; John Gardner, CCBL; John Sayle, CCBL.

Mary McAleese, as director of IPLS, was the courseware commissioner and had the best perception of exactly what the students needed.

Barry Valentine, a Barrister, author of the leading text on Criminal Procedure, and a tutor at IPLS, acted as the script writer and subject matter expert.

Patrick Brannigan was the project leader and as such ultimately responsible for the success of the undertaking. With a background in teaching and computing he was well placed to recognise what could realistically be achieved within the final product.

John Gardner, as an associate director of CCBL and Reader in the School of Education, was able to offer advice on pedagogical aspects and help specify the desired modes of user interaction.

John Sayle was a junior programmer on a year's work experience at CCBL. He started to learn the chosen authoring system as soon as CCBL committed to the project. Under the supervision of the project leader he was able to assist in the production ofprototype screens and supply most of the programming effort required to complete the project.

Courseware Subject Matter

We chose Criminal Procedure (Indictment) for our first venture. Having taught Criminal Law and Procedure for nigh on seventeen years Mary McAleese was aware how great was the gap between the 'LA Law' laden student perception and the reality of plodding through a minefield of rickety rules and regulations which could reduce the most sensational fact situation to something about as exciting as a dentist's waiting room. Yes, Criminal Procedure needed to be glamorised. We divided it into its two natural components -trial on indictment and summary trial - and started with the former, learning from our mistakes when we came to the second program. The former has been in use now since 1990, the latter will be introduced this year.

Criminal Procedure is currently taught each afternoon over six consecutive weeks with a lecture and tutorial or mock trial in each daily session. The amount of material to be covered is considerable. It is dense, detailed, bulky and complex. The scope for pondering over the 'lectured' material is limited. What we needed was some additional opportunity for 'vertical' study. The lectures steamed along 'horizontally' but there was very little opportunity for students to move at their own pace.

Development

Design of CPI

An initial meeting of the project team discussed the project to clarify the possible forms the courseware could take. We 'brainstormed', tossing about a number of ideas, some of which were never used. The subject matter experts were asked to define the educational aims. This provided the

other members of the team with a perception of the target audience and the appropriate context of use.

In this particular project a script for the courseware was provided after only one meeting of the project team. This was contrary to the preferred working practice of CCBL, which is that the script is developed collaboratively to ensure the best final presentation on the computer. However in this case the subject matter was clearly laid out and outlined and the project leader had no difficulty accepting it.

Many options were open as to what software to use on the chosen Macintosh platform. It would have been possible to 'start from scratch' with a programming language such as C, Pascal or even BASIC. However utilising one of the already available authoring systems meant that powerful software tools were immediately available. We adopted HyperCard which was one such system. It is supplied free of charge on every Macintosh and comparable systems were expensive. This system provides a framework (known as a stack) of screens (cards). Each card can contain fields (of text), buttons (special areas of the screen), and graphics. Backgrounds consisting of any of these card elements can be defined and used with other elements on a particular screen. The stack, cards, backgrounds, fields, and buttons are all referred to as objects. Any object can have a corresponding set of commands associated with it (known as a HyperTalk script) and this defines what actions occur upon a particular user interaction with that object. For example the user interaction of clicking the mouse when the pointer is over an object such as a button could result in the action of another card being displayed, perhaps in another stack. For further information the reader is referred to the Apple documentation or any of the many popular books available for HyperCard (e.g. Schafer, 1988; Waite, 1989). The extensive flexibility of such a scheme was more than sufficient to satisfy the demands of this project: the programming consisted of arranging suitable objects in backgrounds and cards and preparing appropriate HyperTalk scripts where necessary.

The project team met to discuss comments which the project leader presented about the subject matter script and to review prototype screens which had been produced with Hypercard. It was useful to have such prototype screens because all team members then got a good idea of what the computer could offer and the subject matter experts were able to contribute more fully to the project.

Within the team the educational aims were challenged and confirmed. Agreement was reached on what assumptions could be made about the students' familiarity with basic computing skills and their likely reaction to elements of a graphical user interface (such as 'buttons'). In this case it was assumed that they had no previous experience of using a computer.

Subsequent discussion about how the package should appear included such general points as: how questions were to be handled on screen; how would 'navigation' between different screens be implemented; what text density was acceptable on the screen; was there a need for continuation screens when questions were long and how would this be introduced. The inclusion of a scoring mechanism was also debated, but we decided not to include it in this package - we weren't convinced that the extra programming effort required would be justified by the return in terms of meaningful scores. However we did modify our views on this topic when we came to consider the next package (MC).

After such meetings, the project leader was able to form a better programming overview of the project: what the main modules would be and how they would interact. The main features of screen handling were also identified.

Production of CPI

The programmer built the Hypercard stacks with regular recourse to the project leader who emphasised issues such as: screen consistency; ease of use; the inclusion of buttons to provide 'pop-

up notes' where certain classes of information might clutter the screen; the use of other buttons to remove pop-up text.

There was continual contact with the subject matter experts to clarify minor details in the material.

There was a need to work on the slower Macintosh Plus model which the students would use, rather than faster models available at CCBL which gave a deceptive impression of machine response.

Special measures had to be incorporated to prevent the naive user from inadvertently destroying some of the program or exiting to 'unchartered territory'!

Even at this stage circumstances arose when it was unclear how to best program parts of the courseware script in HyperCard. For example in one question the student was asked to re-order items; several methods were prototyped and one selected by the project leader. Subsequent project team meetings reviewed and endorsed (or otherwise) such decisions.

With further production and review work, it became obvious that the material should have more structure. There were too many questions in one section. It was thought that the student user would 'get lost' or their attention span would be strained. The script writer delivered the necessary breakdown into further sections.

With more of the courseware available in prototype form, other reviewers were invited to offer opinions. Helpful information was gleaned from this formative evaluation. For example one reviewer had experience of rooms full of computers and drew attention to the potential cacophony of sound coming from all the machines at once: as a result the project cut back on the use of sound although it was still retained for certain screen activities.

A considerable overhead in terms of time was that the material had to be proof read on screen. This was very important to ensure the quality of presentation and had to be done by both the script writer and someone on the production side. In general it is helpful if the material is typed and available electronically: in this case the script was hand written but the writing was clear!

As the package neared completion it was demonstrated at several seminars. Again some adjustments were made as a result of useful comments.

Description of Completed CPI Package

The delivery machine is an Apple Macintosh Plus equipped with external 20 Mb disc. The course subject matter is divided into sections. Each section is associated with a group of cards in a HyperCard stack. A card contains fields of information and buttons for user interaction via a mouse device. The learning activities presented on a card include answering questions and clicking on particular buttons to retrieve further information.

The program offers three plus hours of training material on Indictable Procedure from pre-trial, through trial, to post trial. Figure 1 provides an overview of the content.

Figure 1
CPI Package Content Overview
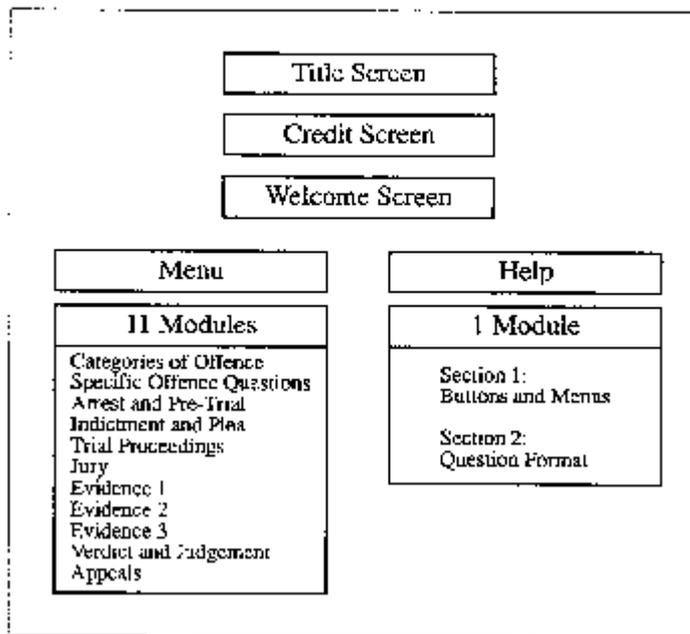
There are twelve modules; one is a Help module, another provides a browse facility for reference material which sets out the categories of criminal offences which exist and gives examples within
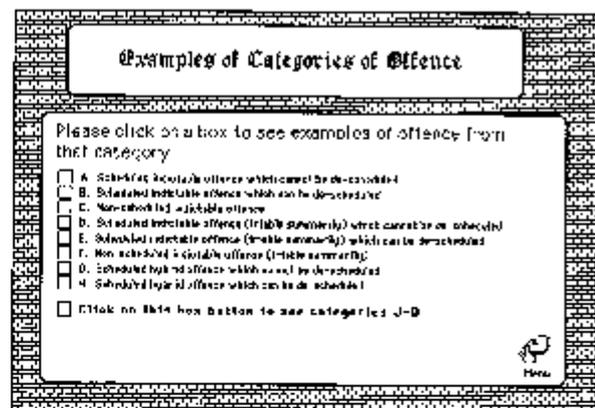
each category. The ten main modules are described in Table 1. ▣

Figure 2 shows sample screens from the package: 2a is the menu screen from where the user selects which module to tackle; 2b is the first screen of the browse module; 2c and 2d are typical question screens.
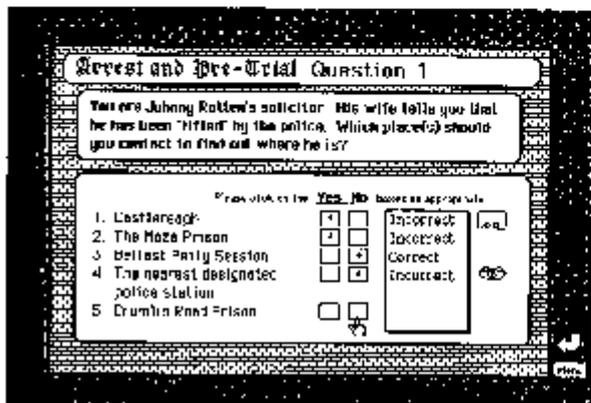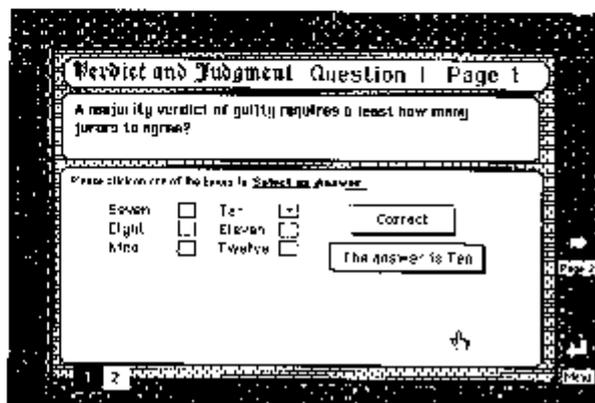
Figure 2
Some Screens from the CPI Package
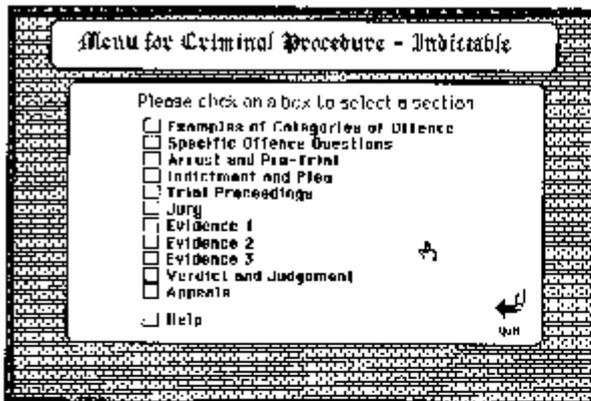
a) The Main Menu Screen

b) First Screen of Categories of Offences Module

c) Questions Requiring Yes/No Selection

d) Question Requiring Selection from Several Options

The package is designed to instil, and reinforce knowledge of quite detailed factual information. It uses a variety of techniques to do so, conscious of the dangers of adopting a rote-learning approach which woulc quickly deaden the impact of the program. Each module or section is self-contained, although part of unified program. The user may be asked to say yes or no - true or false - to a series of assertions (Figure 2c) or he may be asked to identify a correct answer from within a range of given 'answers' (Figure 2d) Answers are entered by clicking on the relevant answer box and feedback is immediate. The user is told his answer was correct or incorrect. Relevant authorities are cited for further study but text is kept to minimum and there is no opportunity for students to enter 'free hand' text.

The MC Package

Much of the ground work for the second package, MC, had already been done with CPI. From the IPLS point of view it was a continuation of the CPI project and a script was furnished as before; the content covered practice and procedure in the Magistrate's Court. George Munroe, CCBL, became the new project leader. New ways of implementing the material with HyperCard were proposed and prototyped. The reaction from reviewers was favourable.

Having studied the completed CPI package, the project leader had determined the following further objectives with regard to the 'look and feel' of the MC package: better use of screen 'real estate'; more screen consistency throughout the package; a more adventurous user interface, which would allow the experienced usermore convenient access without threatening the computerphobics. Figure 3 demonstrates some of these points (a fuller discussion of such issues can be found in e.g. Rivlin etal, 1991; Riley et al, 1990).
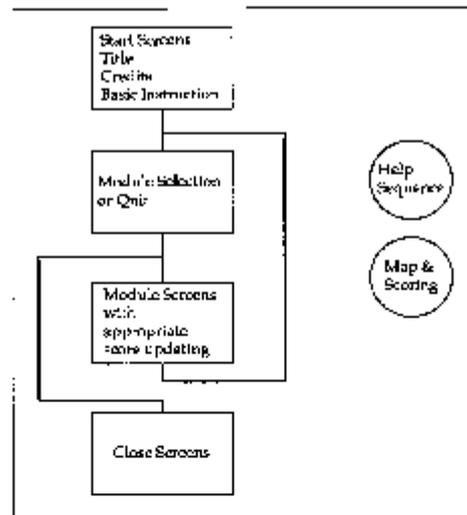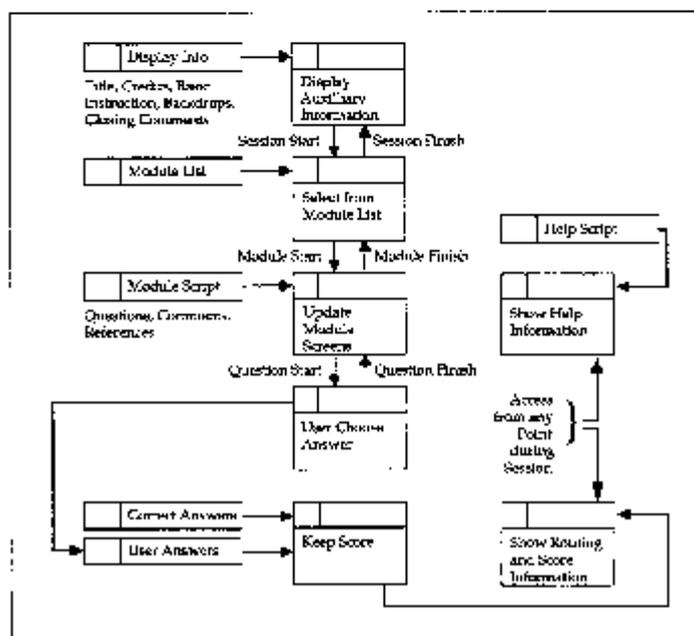
Figure 3
The User Interface in the MC Package
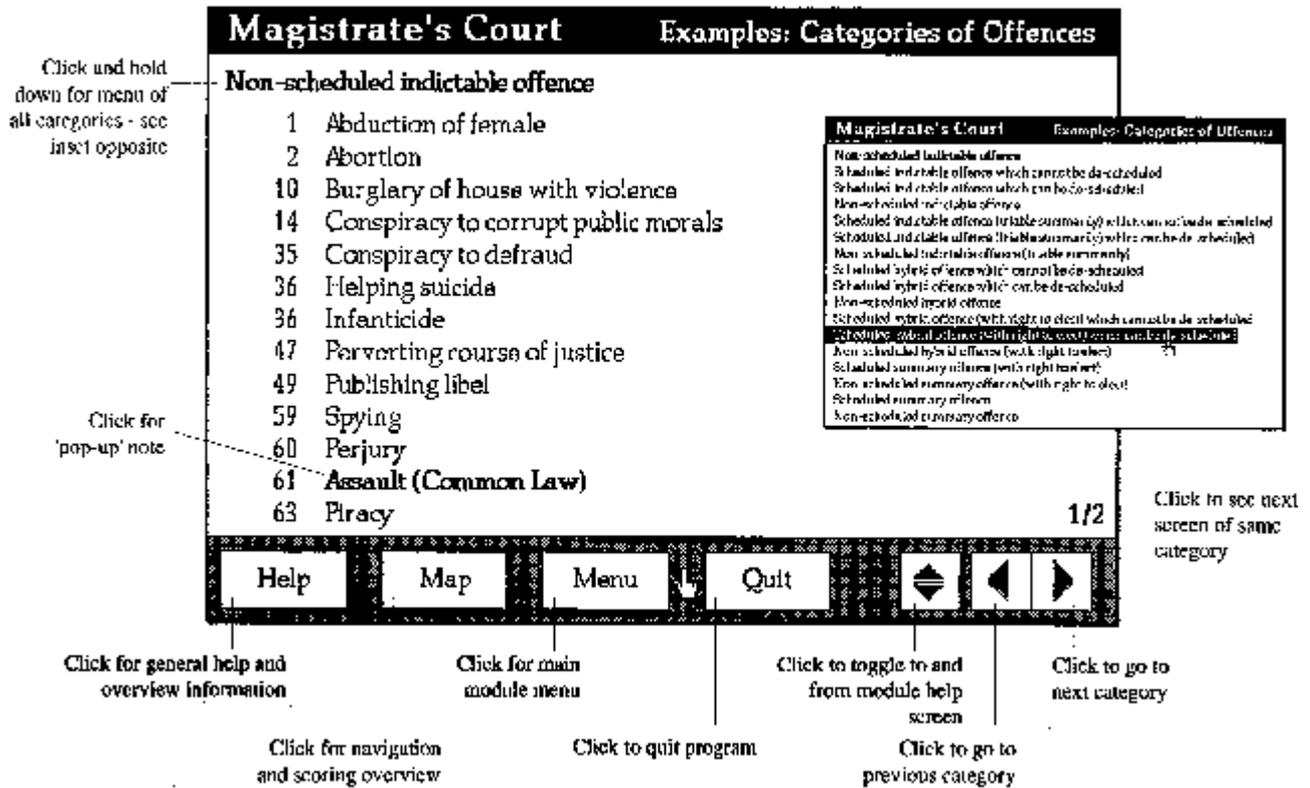
The full screen is used, corner to comer, allowing larger clearer text. Three areas of the screen are clearly distinguishable: the orientation information in the black bar along the top; the main learning activity in the centre of the screen; and the grey bar along the bottom containing all the navigation buttons. The navigation bar remains identical for every screen in the package; the orientation bar's contents change but its position remains the same; the appearance of the centre of the screen changes with different modules. Certain 'rules of engagement' remain in effect for the centre area even though the contents and layout vary. For example by moving the mouse pointer over emboldened text and holding down the mouse button some action is initiated: a pop-up note appears; a list is extended; a drop down menu appears to allow further selection.

As well as 'face-lifting' the user interface, a much more fundamental improvement to the design process was asserted; namely a fuller functional analysis of the required courseware. Reference to Figure 4 may help clarify exactly what we mean by this.

Figure 4
Overviews of the MC Package

a) The User Perspective

b) The Software Engineer Perspective

Figure 4a is a brief user overview of the courseware (the circles on the right simply symbolise facilities that should be available at all times) - the sort of diagram that could be sketched at a meeting of all the project team members to confirm understanding of the basic structure. However, before this can be transformed into reality on a computer screen using programming 'sorcery', a thorough analysis should identify exactly what 'data' or information items are involved and what 'processes' are required to act on these. Figure 4b shows part of one such analysis for MC; the smaller boxes indicate the data stores, the larger boxes the processes, the horizontal arrows represent data flow to or from the stores, and the vertical arrows are user generated 'events' (the full depiction involves the elaboration of most of these 'top level' processes into sub processes). With this interpretation most of the supplied subject matter script finds its way into some of the data stores, as does supplementary information such as help and the user responses. The delineation of data and processes promotes a software solution which allows the easy updating or modification of the subject matter (because this is essentially isolated in passive repositories) and it circumvents unnecessary programming at the production stage (since the same processes can be applied over and over again where necessary in the final program). Such an approach to design and a 'generic' programming attitude (where variable names are used in code rather than actual values) constitute basic software engineering skills. It Is our experience that quality courseware cannot be realised without them.

During the design of MC the advantages of jointly developing the subject matter script became apparent. If the completed script was to be accepted as definitive then some 'messy' programming would have to be employed which would slow down machine response; whereas rearrangement and standardisation of parts of the script would allow much more 'elegant' programming. The synergy between subject material and enabling software, which is required for professional courseware, demands that the expectations of the subject matter expert and the software designer evolve together.

No mechanism exists in CPI for monitoring the student's experience of using the program (although the user can go back and see how they responded within one session). This was a design decision which was adhered to, though misgivings were expressed from time to time. With MC we have decided to opt for such a scoring facility. The decision was informed by several factors: student preference for such a facility; comments made in the evaluation of CPI by a researcher at QUB (see Evaluation of CPI below). Figure 3 shows a 'map' button in the navigation area of the screen. The intention is that upon clicking on this button the user will be given a short report on the number of questions already attempted and how many of these were answered correctly. It is hoped that along with this report a pictorial plan of all the questions in the package can be displayed with indicators as to which ones had not yet been attempted; the user could then click on any question to go to it directly.

Some pictures which have been produced for MC by CCBL's part-time graphics design artist, Wendy Rutherford.It is believed that such graphics will make use of the package even more attractive. For some of the court scenes Wendy attended Belfast Magistrate's Court to gain first hand experience of the setting.

We hope to pilot the MC package in March 1992 and have it ready for full integration into the course schedule in the academic year 1992-93.

Implementation

Use of CPI

The completed CPI package was introduced in June 1990 to 20 Bar students. Each spent three to four hours, working either in groups of two or alone, under the supervision of a tutor who was available to help with problems but did not 'conduct' the session. The feedback was enthusiastic and there was a marked preference expressed for working in groups of two. Since the program posed many highly specific factual questions, students found the joint search for the correct answer to be more productive and instructive, than a sole search. However, while all felt the program was useful and quite different in effectiveness from conventional lectures, its main thrust was seen as a means of 'reinforcing' factual information delivered by the lecture. Interestingly, by the time students came to use the CPI package they had already completed (only a week earlier) the 36 hour lecture and practical course and examination in Criminal Procedure Indictment. Even students who had performed well on that course were tripped up by the detailed questioning of the CPI program. Time and again students remarked on the value of the opportunity to reinforce the 'consumption' of important detail, which had already been thoroughly covered by lectures and written materials but had not been firmly and confidenily committed to memory. A number of students (4) volunteered that the CPI program enhanced their confidence in asserting factual information.

In 1991 we built CPI into the Indictment course, but again for Bar students only. The 70 Solicitor students who also took the Indictment course, took the lectures and practicals only, but not the computer based tutorials. All students took the same examination at the end of the course. Analysis of the examination results produced the following information:

Examination Marks

	Lowest	Highest	Average
20 Bar Students (with CALT)	55	72	65
70 Solicitor Students (without CALT)	45	78	59

The differential of 6% in the average marks is interesting, but far from conclusive. This same group of students took nine examinations together in the following subjects: Accounts, Chancery, Company, Consumer, County Court, Wills, Conveyancing, Summary Procedure. In all but two of these examinations in Consumer and Summary Procedure there was no more than a 1% variation in the average Bar and Solicitor performance. In the other two however there was a 12% and an 8% variation in favour of the Bar. Explanations for these variations are speculative. Did the CPI package benefit the Bar students in examination performance?

Another possible explanation is that the three courses in which they did best (this includes Indictment) were dominated by Barrister tutors rather than Solicitor tutors. This may have oriented the material and expectancy in a way which favoured the Bar students.

However the examination results in Indictment in 1989 show a differential of 1.46% in favour of the Solicitor students. The differential in both Consumer and Summary in 1989 was 1% in favour of Solicitors. The argument about course orientation and impact of the tutors is weakened since these remained the same in both years. The differential in six of the nine subjects taken in 1989 was no more than 1-2% either way except in County Court, Revenue and Wills where 6% differential in all three favoured the Solicitor students. One might have been inclined to suggest that the four months spent by Solicitor students in-office (but not Bar students) prior to admission to the Institute, gave them an edge in those 'Solicitor' oriented subjects, but the argument does not hold good from year to year.

Overall feedback from the students introduced to the program was enthusiastic. They enjoyed handling the mouse, found the program easy to follow and use, would have liked an evaluation or examination facility, as one student said, so that they 'could go back and beat the machine next time'.

In March 1992 the program will be used by all 90 student for the first time. We are no longer tied exclusively to the three 'in-house' Macintoshes and will for the first time be making use of a 'bank' of open access machines in the Computer Science department, to conduct the Indictment 'seminar' simultaneously for all 90 students, each of whom will spend four to five hours over two afternoons on the package. The cost savings on tutors for these sessions will be the first opportunity CALT has had to 'pay back' the initial modest investment.

Evaluation of CPI

Evaluation has taken three forms: 'anecdotal' feedback from associates; a formal assessment of the SWIG work; and a comprehensive evaluation using the MEDA tool. The MEDA evaluation tool was developed bya consortium of European groups from teaching and industry specifically for evaluating training software (Machell and Saunders, 1991).

Anecdotal feedback from tutors and students was not very scientific but, far from discouraging, it contained incentive to keep developing and some ideas for improvement.

An assessment of the SWIG initiative (Student Workstation Programme as it is referred to in the official report) was carried out for the Department of Education for Northern Ireland by John Gardner and Helena McNally (School of Education, QUB) in 1991. Their report (Gardner and McNally, 1991) comments on the overall effectiveness of the SWIG initiative rather than on individual projects conducted under its auspices. However the CPI package was filmed in operation, and students and tutors were interviewed. Student and staff response to the program was

unanimously enthusiastic. Across the University staff response to SWIG was also unanimously enthusiastic and approving while 85% of students interviewed had their attitude to computing influenced for the better. The report comments favourably on the significant degree of self-help and peer assistance noted in the classes which were videotaped. Tutors played an interventionist role only 'on demand' but were freely sought for advice as need arose. Those who teach tutorials and who find it difficult at times to promote student discussion maybe heartened by this feature of computerbased tutorials.

Evaluation by Bernadette Gaffney, a solicitor and researcher with the Faculty of Law at QUB, carried out in 1991 on the CPI package, using the MEDA evaluation tool, contained the following main comments:

i) The quality of user-program interaction is good.

ii) With modification it could be of use to other persons involved with the criminal process, e.g. police, court service, legal executives, probation officers, social workers.

iii) The screens throughout are consistent, clear and amazingly text light when one considers the amount of dense legal information to be presented.

iv) The program is easy to operate and instructions for use are clear and exhaustive - a trainee with few computer skills can use it unaided.

v) The program does not provide a way of evaluating what trainees have retained from the instruction process. The effectiveness of the question and answer strategy is lessened by this omission.

vi) The design of CPI is impressive for its clarity, appropriateness to subject content and user friendliness.

vii)A precise statement of learning objectives could have been included at the beginning of the program and for each module.

viii) CPI is an innovative, interesting, motivating approach to the teaching of law.

Discussion

In this University deliverable courseware is a reality because of the existence of CCBL working with discipline specific departments, such as IPLS. Without such a unit to supply expertise, in what is essentially a new field, the development of any computer based teaching packages would be difficult: the market prices for external contract work would be prohibitively expensive; the resource demands on a subject matter enthusiast would be substantial.

We have estimated that the cost, in time, of the CPI package development was around 650 hours. Most of this was spent by the student programmer (480), then project leader (135) and script writer (24). The financial cost to IPLS amounted to funding a student programmer at CCBL for six months. We expect that the cost, in time, of the MC package development will be a fraction of that of CPI.

We believe that 'good' working practice in courseware development embraces the project approach with the forming of a team of individuals who can between them provide the necessary range of skills - subject intimacy, writing (and perhaps drawing), screen presentation, user empathy, software analysis and design, programming. These latter two encompass the ability to identify the fundamental software requirements for transforming the subject material into a computer package.

The project should progress with systematic prototyping and reviewing.

A major problem occurred when CPI was first used by students which highlighted an oversight within the project. Student interference (usually accidental, but occasionally deliberate) on the computer during a session prior to running the CPI package was resulting in undesirable effects, ranging from trashing the CPI package to corrupting the whole system setup. During development the preoccupation had been to ensure fail-safe operation within the package. The problem was addressed and resolved by arranging semiautomatic execution of the package on machine startup and protecting all 'sensitive' files on the machine against inadvertent or malicious abuse. For its duration the problem was a source of frustration and a waste of valuable time.

The availability of support for courseware is important. In the case of the projects described in this paper CCBL has been 'on call' to resolve problems, make adjustments to the subject material, and provide upgrades to a better system (Hypercard version 2).

The HyperCard environment can appear a deceptively easy system to exploit. However in order to create courseware which is well structured, self consistent, and easy to maintain, it is necessary to enforce a design discipline within HyperCard. HyperTalk script handlers (which would correspond to the processes of the software engineer's perspective as in Figure 4b) should be installed at as high a level in the object hierarchy as possible (preferably in the stack script, then background, and less frequenfly in the card script).

We need to develop a policy for the future which will allow us to maximise the audience for our computer based packages as they are developed, without losing sight of their main purpose as a teaching tool for trainee lawyers. With open access to computers in libraries (public and university) within sight, we hope to offer our programs for public access, and promote their use among personnel in the Criminal Justice field. However there is also a need for a campaign of proselytisation amongst our colleagues, many of whom are not yet convinced of the worth of CALT.

We are satisfied that the aims of the computer assisted learning initiative are being fulfilled and the new opportunities for students to pursue self paced interesting tutorial sessions without supervision has made our projects very worthwhile.

The next project, after completion of Magistrate's Court, is an interactive video program for Advocacy training. Preliminary discussions have already taken place but costing and timetabling are still to be finalised.

References

Apple Computer Inc.; Human Interface Guidelines: The Apple Desktop Interface (Addison-Wesley Publishing Company); 1987

Apple Computer Inc.; HyperCard User's Guide (published by Apple and supplied with new Macintosh models up to June 1991)

Apple Computer Inc.; HyperTalk Beginner's Guide: An Introduction to Scripting (published by Apple and supplied with new Macintosh models up to June 1991)

Schafer D; HyperTalk Programming (Hayden Books, Indianapolis); 1988

Waite Group (M Waite, S Calamar, B Barrett, G Jorgensen); Tricks of the HyperTalk Masters (Hayden Books, Indianapolis); 1989

Rivlin C, R Lewis, R Davies-Cooper; Guidelines for Screen Design (Blackwell Scienflflc Publications); 1990

Riley D, R Millwood, A Edis, M Figg, D Riddle; Design for Active Learning with HyperCard (published by the Centre for Educational Studies, King's College London); 1990

Machell J, M Saunders; Evaluating Training Software - MEDA (a CSET publication, University of Lancaster); 1991

Gardner J, H McNally; An Evaluation of the Impact of the Student Workstation Programme at The Queen's University of Belfast (published by the School of Education, The Queen's University of Belfast); 1991

Trademark Acknowledgements

Hypercard, HyperTalk, and Macintosh are trademarks of Apple Computer, Incorporated.

Authors

Mary McAleese, LLB, MA, Barrister-at-Law, is Director of the Institute of Professional Legal Studies (IPLS) at the Queen's University of Belfast.

Patrick Brannigan, BSSc, PGCE, MSc, and George Munroe, BSc, PhD, are staff of the Centre for Computer Based Learning (CCBL) at the Queen's University of Belfast.