A SELF-TAUGHT COMPUTER ENGINEERING COURSE

J.N. Coleman, F.P. Burns, D.J. Kinniment, Dept Electrical and Electronic Engineering;
T.J. Butler, Dept Medical Statistics;
A.M. Koelmans, Dept Computer Science;
University of Newcastle upon Tyne, U.K..

ABSTRACT

The application of computer-aided-learning as a direct replacement for, as opposed to an adjunct to lecturing, is still in its infancy in higher education. This paper examines some of the reasons for its slow uptake and then describes a project to develop courseware for a large proportion of the Electronic Engineering syllabus within several U.K. Higher Education Institutions. The first modules to be completed cover the area of Computer Engineering. We describe the philosophy and design of this courseware, and then report a series of tests in which the examination performance of students using it was compared with that of control groups taught in traditional lectures. The results clearly suggest that carefully-designed courseware can lead to a large reduction in teaching time, with no significant difference in learning.

1. INTRODUCTION

In recent years a great many computer-based packages have entered the higher education curriculum. Examples of interest in the field of Microprocessor teaching include register level simulators [1], and higher-level simulators such as [2]. These educational packages have in common that they are intended to augment a traditional lecture-based course, in order better to illustrate the principal theme of the syllabus. They are not, by and large, intended as a direct replacement for the lecturing.

Observers have drawn attention to the fact that the application of such 'courseware', whilst becoming increasingly accepted in secondary schools, is still in its infancy in higher education [3]. Reasons postulated include the lack of suitable courseware, lack of time and staff to develop new material, and lack of financial and administrative support to do so. Whilst these problems are obviously soluble, a number of more fundamental doubts about the desirability of this type of teaching have tended to deter university staff from making the necessary commitment to it.

Let us examine some of these objections in detail. The first concerns its effectiveness as a teaching technique. Do the students learn as well from it as from traditional lectures? There is as yet little objective evidence on this point.

Related to this is the fact that students in higher education are much less tightly supervised than their high-school counterparts, and a greater degree of self-discipline is therefore required of them if self-teaching is to be a success. Is it reasonable to expect sufficient commitment from them to make the experiment work? A second issue derives from the high degree of specialisation of university-level courses, and the tight integration of the several courses comprising a particular degree. This tends to prevent the easy transfer of courseware from one institution to another, and thereby exacerbates the shortage of suitable material.

Against these objections, a number of benefits have been suggested, some indisputable and others open to question. The one certain benefit is the reduction in staff lecturing time. However, it would be hoped that the time spent by the students in their self-teaching activities would be less than that taken by the original lectures, yielding a saving in their time too. Another possible advantage is that the students may positively welcome the new teaching techniques, and feel more motivated to achieve better results.

In 1992, starting from a base in which almost no suitable material was available, a collaborative project was established between several U.K. E.E. and C.S. departments with the aim of developing courseware for a large proportion of the identifiably common parts of the degree syllabus. In doing so, we had an opportunity to investigate the questions posed above. Standing midway between the creative arts and natural sciences, engineering was a suitable discipline for this experiment because of its reliance on large quantities of diagrammatic information. Whilst it was not felt that CAL could ever portray the creative design concepts pervading any advanced engineering activity, it could possibly express many of the fundamental techniques, ideas and principles characterising the early stages of a degree education.

By 1994 the first piece of courseware was ready for use. It was entitled 'Computer Architecture', and mounted on PC / Windows systems. During the second term of the 1993-94 academic year it was tested by dividing a class of Electrical Engineering students; half doing the courseware and half coming to the equivalent lectures. At the end of the trial, a short test was given to the entire class to see if any difference could be observed in the scores obtained by the two halves. A more detailed analysis was then made, which took into account their overall average for the year. Encouragingly, we found that there was no significant difference between the two halves, but we did find evidence

to suggest that the CAL may actually be superior to the lecturing in bringing weaker students up to standard. The remainder of this paper describes the courseware itself, the experimental procedure, and the results we obtained. It then describes the integration of this and further modules into a complete self-taught Computer Engineering course, due for use in 1996-7.

2. THE COURSEWARE

Within the engineering disciplines, a degree course typically includes a series of specialised lecture modules of approximately one term's duration. The modules exhibit a complex pattern of interaction; some progressing sequentially, others reinforcing concurrent modules, and others having an all-pervading influence throughout the course. It is often a matter of some difficulty for the lecturing staff of the department, as they struggle to keep abreast of a rapidly changing subject, to keep the modules within a coherent framework. The resulting courses vary quite markedly between institutions, reflecting the different perceptions of priorities at each.

From the outset then, it is unlikely that a single suite of courseware could be appropriate for all degree courses in the subject. It is also unlikely that sufficient variants could be constructed to satisfy all the requirements. The designers of the courseware sought to bypass this problem by adopting a strategy of very fine-grained modularity. The material would be divided into very small modules, each corresponding to only a few hours of lectures; very much smaller than the one-term granularity of a lecture-based course. At this level of resolution less difference is observed between the teaching at separate institutions. Where a lecturer was satisfied that part of his material was adequately covered by the courseware, he would be able to substitute this for the lectures. Where his own material differed from the courseware, he would continue lecturing as before.

We were not unduly concerned that a patchwork of lectures and courseware might develop; indeed, we would seek to augment the variation by introducing practical work and written exercises. The aim was to counteract the inherent dullness of sitting at a computer screen for long periods by frequent changes of activity. We were somewhat anxious about this point because of the minimal supervision of the students, and the high probability that, left completely unsupervised, the less well-motivated may be tempted to hurry through the computer-based material with little attention.

Thus the syllabus was divided into themes, and each theme into a set of modules. The first theme to be addressed covered the area of digital electronics. This was divided into 17 modules, of which the first to be completed comprised a basic introduction to the operation of a CPU. Entitled 'Computer Architecture', the module corresponded to four hours of lectures, and was used as an experiment to evaluate the effectiveness of this type of teaching.

The material we were dealing with was detailed and factual in nature, and not intuitively appealing to many students. A number of techniques which had been successfully employed in the original lecture course were also used in the courseware to maximise the chances that the latter would work successfully with minimal supervision. One important factor was the use, as far as possible, of an inductive style of reasoning. Thus we proceeded by prompting the student to recall some well-understood fact, then drew an obvious conclusion about the principle we were trying to illustrate. Often we used an explicit question-and-answer format in consecutive frames, for example:

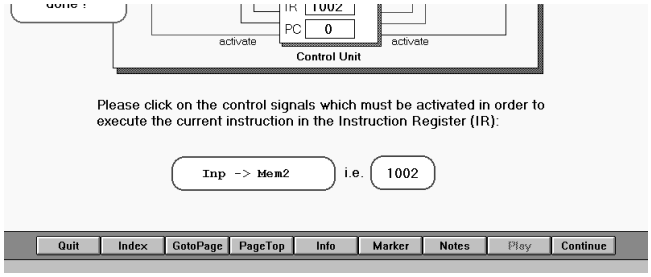
Question :	What does a computer do?
Answer:	It inputs information, stores it internally, processes it, then outputs the results.
Conclusion:	So a computer must have four blocks of electronic circuitry: to input, store, process and output the information respectively. They must be connected so as to be able to pass information to each other.

This leads directly to the basic model of a computer system comprising input, memory, arithmetic unit and output; a model which is used to inculcate the concept of processing throughout the early part of this module. It is accompanied by animated displays in which the concept of a machine instruction is introduced naturally by showing data moving around the system in response to simple high-level-language commands. The student is then prompted to realise that the machine instructions themselves must have some physical embodiment, and from this is developed the idea of the fetch-execute cycle, control unit, and attendant hardware items such as the instruction register and program counter.

As the complexity increases, a more tutorial style is unavoidably used to present the less intuitive aspects; for example the student is directly told how the hardware units are organised internally. In any case, a deliberate attempt is made to juxtapose contrasting styles of presentation, another commonly-used technique being example-driven.

High degrees of animation and interaction are of crucial importance to the presentation. Animation is a particularly valuable technique when it illustrates a point which cannot effectively be described with a static lecture slide. Interaction is used throughout to maintain the students' interest and to act as a check on too rapid progression through the material without proper understanding. An example is shown in fig. 1. Here the student is asked to point to the electrical signals which will be activated as a particular instruction is executed. When the correct signals

are identified, the student is rewarded with a moving display of the execution of this instruction. A development package, Authorware Professional, was specially chosen for its well-developed animation facilities. It offers precise control over visual composition, timing, and trajectories of moving objects, and allows flowchart-based sequencing controlled by accurate positioning of the on-screen cursor via the mouse.



Offending Command = image
Error = nametype : rangecheck

Stack =

--nostringval--
øøøøøøøøddÿÿÿ